

# **CORDIS-ANIMA SYSTÈME DE MODÉLISATION ET DE SIMULATION D'INSTRUMENTS ET D'OBJETS PHYSIQUES POUR LA CRÉATION MUSICALE ET L'IMAGE ANIMÉE**

**C. CADOZ, A. LUCIANI, J.L. FLORENS**

ACROE - LIFIA  
Institut IMAG, 46, Av. Felix Viallet 38000 GRENOBLE - France

## **RESUME**

Dans cette conférence, nous présentons CORDIS-ANIMA.

*CORDIS-ANIMA est le nom d'un système pour la modélisation et la simulation numérique en temps-réel, d'objets physiques manipulables, audibles et visibles. Développé et expérimenté par étapes successives depuis 1979, il constitue le noyau d'un outil informatique pour la création musicale et l'animation d'images. CORDIS est plus particulièrement dédié à la création sonore, ANIMA à la création du mouvement dans l'image, mais l'un et l'autre ont une très forte base conceptuelle et matérielle en commun.*

Dans une première partie, en nous appuyant exclusivement sur le fait que la simulation d'un objet physique à l'aide d'un système électronique représente toute communication mécanique par des liaisons ponctuelles, uni- ou multi-dimensionnelles et orientées, nous montrons quelle est la forme générale nécessaire d'un objet simulé. Adoptant alors comme principe qu'un langage de modélisation et simulation des objets physiques doit permettre de construire les simulacres à partir d'un nombre minimal d'éléments et que ces éléments doivent être de même nature que le tout auquel ils participent, nous en déduisons les principes et la forme théorique générale de la modularité dans la modélisation et la simulation physiques.

Aux modules physiques dont la vocation est d'établir un lien avec les objets du monde réel, le langage CORDIS-ANIMA associe des modules fonctionnels dont le rôle est fondamental pour l'ouverture du système, l'extension, l'extrapolation et l'intégration de fonctions non physiques très générales compatibles avec le formalisme de construction de base.

Dans une deuxième partie, nous envisageons les conséquences du fait que toute variable continue est représentée, dans un contexte de calcul avec l'ordinateur, par des signaux échantillonnés répondant aux conditions de Shannon, et que tout module de calcul (ou module de simulation) calcule en un temps fini non nul des variables de sortie à partir de variables d'entrée. Nous introduisons la notion de *T-simulabilité*, condition pour qu'un objet physique soit simulable par une machine (réelle).

Enfin, dans une troisième partie, après nous être intéressés à la "forme" générale de CORDIS-ANIMA, nous nous intéressons à sa "matière", c'est-à-dire au remplissage de ce cadre formel par des algorithmes effectifs de simulation d'objets physiques réels et identifiés.

## ABSTRACT

*In this conference, we will present CORDIS-ANIMA.*

*CORDIS-ANIMA is the name of a system for the modelisation and real-time numerical simulation of manipulable, audible, and visible physical objects. It has been developed in successive stages since 1979 and constitutes the nucleus of a computer tool for musical creation and image animation. CORDIS is particularly suited to sound creation, and ANIMA to creating movement in images, but both have a very strong common material and conceptual basis.*

*We will begin by showing the necessary general form of a simulated object, by basing our remarks exclusively on the fact that the simulation of a physical object by means of an electronic system represents all mechanical communication by punctual, uni- or multi-dimensional, oriented chains. By then adopting the principle that a physical object modelisation and simulation language must enable the construction of simulacrum from a minimal number of elements, and that these elements must be of the same nature as the whole to which they belong, we can then deduce the general theoretical form and principles of modularity in physical modelisation and simulation.*

*The aim of physical modules is to establish a link between objects in the real world. The CORDIS-ANIMA language associates functional modules to the former. The role of these functional modules is fundamental for an open-ended system, and for the extension, the extrapolation and the integration of very general non-physical functions that are compatible with the basic construction formalism.*

*Following this, we shall consider the consequences of the fact that any continuous variable is represented, in the context of calculus with the computer, by sampled signals corresponding to Shannon conditions, and that any calculus module, (or simulation module), calculates the output variables from the input variables in non-zero finite time. We shall then introduce the notion of T-simulability. This is the condition that determines if a physical object is simulable by a (real) machine.*

*After examining the general "form" of CORDIS-ANIMA, the third part of our talk will conclude with a close look at its "matter", i.e. how this formal framework is to be filled by effective simulation algorithms of real and identified physical objects.*

## INTRODUCTION

CORDIS-ANIMA est un système de *modélisation* et de *simulation numérique en temps-réel*, d'objets physiques *audibles, visibles et manipulables*. Mis en œuvre et expérimenté par étapes successives à l'ACROE depuis 1978, il constitue le noyau d'un outil informatique pour la création musicale et la création d'images animées. CORDIS est dédié à la première, ANIMA à la seconde, mais les deux ont une base conceptuelle et matérielle commune importante.

La fonction globale de CORDIS-ANIMA, au centre de cet outil de création, est de permettre l'élaboration d'une réplique simulée à l'aide de l'ordinateur, de certains objets du monde physique. Ces objets sont les instruments de musique, plus généralement les objets capables de vibrations sonores sous l'effet de certaines de nos actions, et les objets mobiles, déformables, manipulables tels qu'il peut s'en présenter à notre vision.

La réplique simulée - le simulacre - de ces objets est réalisée à l'aide d'une machine numérique convenablement programmée capable de communiquer avec l'opérateur humain par l'intermédiaire de transducteurs selon trois canaux sensoriels : acoustique, visuel et gestuel. Le canal gestuel est considéré selon ses deux directions : de l'opérateur vers l'objet (action gestuelle) et de l'objet vers l'opérateur (perception tactile et kinesthésique). L'objectif poursuivi est celui d'une simulation *totale* c'est-à-dire restituant non seulement les aspects sensori-moteurs (gestuels, tactiles, acoustiques, visuels) mais aussi le comportement même de l'objet. Il ne s'agit pas seulement d'une synthèse des phénomènes produits par l'objet, mais d'une synthèse de l'objet lui-même.

Les raisons de cet objectif, bien que fondamentales par rapport au propos principal - celui de l'informatique utilisée pour la création artistique - ne seront pas discutées ici, nous renvoyons à nos articles précédents sur cette question (voir bibliographie ACROE [22], [14], et aussi [8], [15]).

### Un système de modélisation

Deux univers sont en vis-à-vis : celui des objets physiques particuliers évoqués plus haut, disons "l'univers instrumental", et celui des objets simulés. Le premier existe indépendamment de la connaissance que l'on peut en avoir. Pour passer aux seconds il faut acquérir des premiers une connaissance particulière (celle de leurs attributs et propriétés qui ont une incidence sur le processus de production et de contrôle des phénomènes sensibles), et il faut que cette connaissance permette de construire les programmes qui réaliseront la simulation. Un intermédiaire est donc nécessaire, permettant de caractériser et décrire la connaissance utile. Cet intermédiaire est ce que nous appelons ici un modèle. CORDIS-ANIMA joue cette première fonction : celle de nous permettre d'élaborer ce modèle, de *modéliser l'univers instrumental*.

## Un système de simulation

L'ensemble des éléments de CORDIS-ANIMA recouvre un ensemble de modules et de fonctions qui, tout en étant les éléments mêmes de la modélisation, sont aussi ceux de la simulation. En d'autres termes, CORDIS-ANIMA est aussi un système d'algorithmes élémentaires de simulation.

## Un système

Remarquons alors l'importance de cette relation dialectique entre modélisation et simulation : le formalisme CORDIS-ANIMA se distingue des mathématiques de la physique (mécanique) en ce que ses éléments et ses règles sont construits à partir des données de l'univers des processus et processeurs numériques. Mais il se distingue aussi de l'algorithmique de simulation stricte en ce que ces éléments et ces règles répondent de fonctionnalités propres aux objets physiques et à l'objectif de création, formant un système en soi. Les algorithmes effectifs sont des réalisations particulières de ces fonctionnalités, contingentes de l'état de la technologie et de l'expertise.

## Un langage

En tant qu'ensemble d'éléments et règles de combinaison, CORDIS-ANIMA est un langage. Il permet de décrire des objets du monde physique, mais également de les reconstruire dans l'espace virtuel de simulation. Plus loin, il permet de construire dans cet espace virtuel des objets qui n'ont pas d'original réel. CORDIS-ANIMA est donc, comme tout langage, un moyen de création.

Dans la première partie du développement qui suit nous présentons le formalisme général de CORDIS-ANIMA. Il résulte de deux impératifs initiaux. Le premier est matériel, inhérent à la technologie : c'est la nécessaire discrétisation et finitude des communications entre l'ordinateur et l'univers réel. Le second est un choix : c'est celui de la *modularité* et surtout de l'*expérimentabilité* ou de la "*physicalité*" des modules. Nous expliquerons ce dernier terme lors de son introduction.

La seconde partie présentera les aspects algorithmiques de CORDIS-ANIMA. Nous entendons par là ce qui, à l'intérieur du cadre formel défini dans la première partie, correspond à une instanciation des modules en des fonctions physiques effectives réalisables par des algorithmes numériques. En raison du contexte de cet article, nous développerons plus particulièrement les aspects relatifs à la production sonore.

Au début de cette deuxième partie, on évoquera une autre contrainte inhérente à l'ordinateur : le temps de calcul. Dans la simulation physique en temps-réel, en effet, le temps d'exécution d'un algorithme est une donnée de sa réalité fonctionnelle. Sans en développer toutes les conséquences, nous définirons la notion de *T-simulabilité* d'un objet physique.



## PREMIERE PARTIE

### Le formalisme général de CORDIS-ANIMA

## I - LES MODULES PHYSIQUES

### 1 - Discrétisation et finitude des communications

Les points de communication de type M et de type L

La simulation d'un objet physique à l'aide de l'ordinateur s'effectue à l'aide d'un processus de calcul numérique. Ce processus n'a d'utilité que si l'on peut, par des dispositifs appropriés, le plonger dans l'univers des phénomènes sensibles et le rendre ainsi accessible à nos actions et nos perceptions.

La technologie même de ces dispositifs, les *transducteurs*, introduit alors une réduction fondamentale. Les champs de forces, de déformations, de déplacements caractéristiques des actions mécaniques réciproques entre nos organes de manipulation et les objets sont réduits à ce qui peut seul transiter par les lignes d'entrée et de sortie électriques : des signaux, uni- ou éventuellement multi-dimensionnels, en nombre limité. *La communication physique entre un opérateur et un objet simulé par un ordinateur est discrète et de dimension finie.* Bien qu'entre le doigt et la touche, par exemple, le contact s'établisse selon une certaine surface, tout se passe comme si ce contact était ponctuel. On doit alors adopter une représentation idéalisée de la communication physique entre objets sous forme d'une communication mécanique *ponctuelle*.

Egalement inhérente à la technologie, une autre contrainte apparaît dans le processus de communication. La liaison entre deux objets physiques est essentiellement non orientée et y sont en cause deux variables duales, l'une intensive (la force, la pression,...) l'autre extensive (le déplacement, la vitesse, la déformation,...). Or, du fait de l'exclusion mutuelle de la fonction capteur et de la fonction effecteur on ne peut parler que de communications unilatérales : d'entrées et de sorties. Cette contrainte se traduit alors par l'obligation de représenter toute communication physique, intrinsèquement non orientée, par une communication bi-directionnelle réalisée par des paires indissociables d'entrées/sorties. Et dans la suite logique de cette nécessité, vient celle de séparer les variables duales et de les affecter respectivement à l'un et à l'autre des éléments de la paire.

En conséquence, à la discrétisation de la communication s'ajoute cette particularité incontournable : si l'on appelle point de communication, pour l'objet ou pour l'opérateur, la paire d'entrées/sorties qui réalise la communication, il existe deux types de points de communication : ceux pour lesquels la variable intensive constitue l'entrée et la variable extensive la sortie, et ceux pour lesquels c'est l'inverse. Dans la mesure où cette circonstance est indissociable de la simulation à l'aide des dispositifs électroniques et où elle est une des bases du formalisme de CORDIS-ANIMA, nous lui avons attaché une terminologie : nous désignons les points de communication du premier type "POINTS M", ceux du second type "POINTS L" (Figure 1).

Enfin, de façon à favoriser une représentation (mentale) situant les objets dans un espace réaliste, localisable et de taille concevable, nous avons choisi de prendre comme référence pour la variable extensive les *positions* et les *déplacements* plutôt que les vitesses. La variable intensive est la *force*.

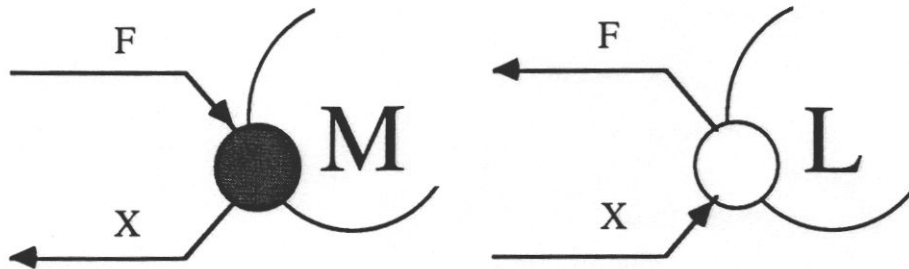


Figure 1

## 2 - Forme générale d'un objet simulé

Ainsi pouvons nous donner comme premier cadre au système CORDIS-ANIMA, ce schéma qui représente la forme générale d'un objet simulé (Figure 2).

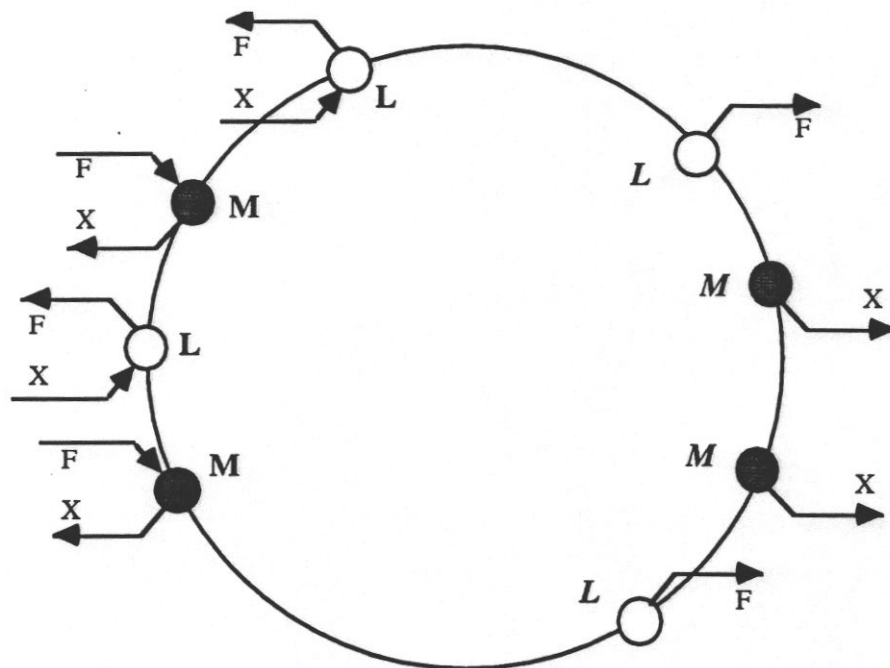


Figure 2

*Remarques :*

- L'entité ci-dessus n'est en fait que la partie "électrique" de l'objet simulé. Nous avons en effet idéalisé l'objet en réduisant momentanément les transducteurs à leur fonction exclusivement électrique. En réalité, ces transducteurs, qu'il s'agisse du canal gestuel, acoustique ou visuel, ont leurs propriétés intrinsèques d'objets physiques réels qui viennent s'ajouter à celles de la partie simulée.
- Le canal acoustique et le canal visuel, bien que physiques ne sont pas mécaniques et en particulier peuvent parfaitement se satisfaire de l'unidirectionnalité de la communication : ce sont des *sorties* pures de l'objet. Il nous a paru néanmoins pertinent de privilégier la vision bilatérale des communications et de considérer que les sorties sonores et visuelles se font par des points de communication de type M ou L, mais "dégénérés", c'est-à-dire sans retour.

### **3 - CORDIS-ANIMA : Modularité "Physicalité" des modules - Modules élémentaires**

#### *Modularité*

Un objet de CORDIS-ANIMA est en général constitué de sous-objets agencés selon une certaine structure. Les propriétés de l'objet global sont déterminées par la structure des relations entre ces composants et par les propriétés intrinsèques de chacun d'eux. Cette modularité est la condition même d'une certaine généricité, c'est-à-dire de la possibilité de construire une grande variété d'objets à partir d'un ensemble limité de composants différents.

#### *"Physicalité" des modules*

Le fondement même du système CORDIS-ANIMA tient dans la clause que nous allons énoncer maintenant :

- Les modules (les composants) d'un objet CORDIS-ANIMA sont de même nature que l'objet global. Ils doivent comme lui être accessibles à nos sens et à nos actions gestuelles. C'est là ce que nous entendons par "physicalité" des modules. Cette clause signifie que tous les éléments auxquels l'utilisateur a affaire, même s'ils ne correspondent à aucun objet existant, ont une réalité physique expérimentable. Leur première connaissance est perceptuelle, sensori-motrice, ce qui n'exclut d'ailleurs pas leur représentation conceptuelle formalisée.

En conséquence, les modules de CORDIS-ANIMA se caractérisent à leur tour par une série de points de communication, de type M ou de type L. Un objet global est alors un ensemble de sous-objets entre lesquels sont établies des communications spécifiques de même nature qu'entre l'opérateur et l'objet global.

De la clause précédente et du principe des points de communication se déduisent les règles de construction structurelle d'un objet à partir de sous-objets. Construire un objet, c'est prendre un certain nombre de composants et établir des communications entre eux en *connectant* leur points de communication. Connecter des points de communication, c'est prendre la sortie de l'un pour en faire l'entrée de l'autre et réciproquement. Les règles de connection se déduisent de la cohérence physique que cela recouvre : l'entrée de force d'un point de type M peut recevoir les sorties de force de plusieurs points L différents. Elle sera la somme (la résultante) de ces différentes forces. Par contre, l'entrée de position d'un point de type L ne peut provenir de plusieurs sorties de positions de point M différents. Il y aurait conflit entre ces différentes positions et il n'existe aucune façon de les combiner qui aurait un sens physique.

On peut alors édicter ces règles de construction (Figure 3) :

- 1) La connexion entre deux points consiste à utiliser la sortie de l'un comme entrée de l'autre et réciproquement. En conséquence, ne sont connectables que des points de types opposés : un point M avec un point L.
- 2) On peut connecter plusieurs points L à un même point M. La force entrante dans le point M est la somme des forces sortantes des divers points L connectés à ce point M. La position entrante dans les divers points L est la position sortante du point M commun auquel ils sont connectés.
- 3) Il est interdit de connecter plusieurs points M à un même point L.

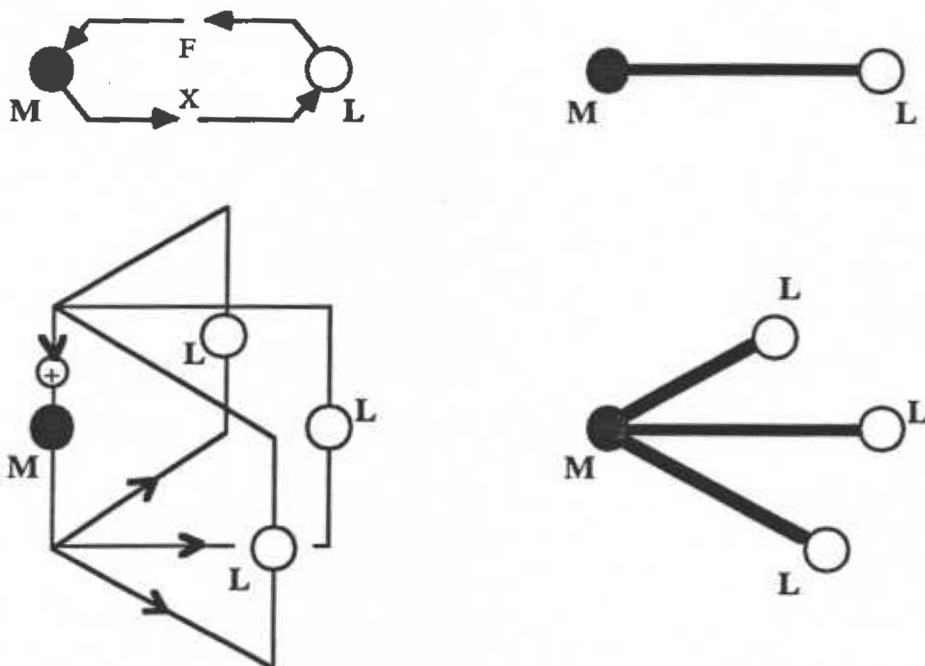


Figure 3

Ainsi, la structure générale d'un objet de CORDIS-ANIMA est-elle un réseau (Figure 4). On peut noter que toute l'information relative à cette structure est contenue dans la désignation, pour chaque point L, du point M auquel il est connecté.

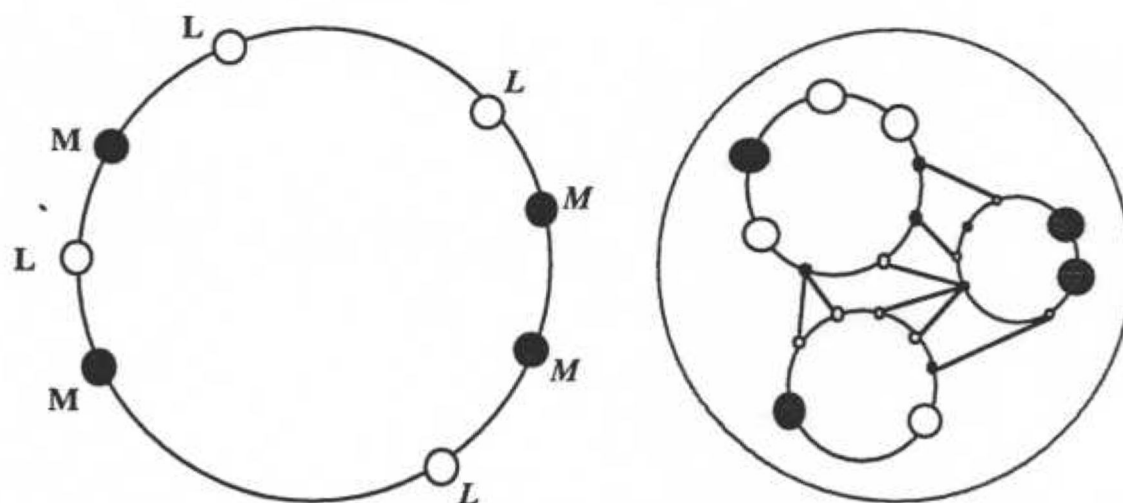


Figure 4

*Remarque :*

Les variables circulant par les points de communication peuvent être uni- ou multi-dimensionnelles : elles peuvent être les coordonnées de points ou les composantes de forces, sur un axe, un plan ou dans l'espace.

*L'Objet interne, l'objet complet. Intégration des transducteurs dans le formalisme*

La forme générale de l'objet simulé présentée précédemment ne correspond, on l'a fait remarquer plus haut, qu'à la partie purement électrique de la simulation. Or, la simulation n'est pas complète si l'objet n'est pas plongé dans notre univers sensible, c'est-à-dire sans les transducteurs. Les transducteurs ne sont pas parfaits, c'est-à-dire qu'ils ne se contentent pas de "transduire". La matière même qui supporte leur fonction a ses propriétés et ses comportements propres. Tout se passe en fait comme s'ils étaient des composants particuliers aux côtés des composants simulés, à la différence près qu'ils sont réels, non programmables. On peut alors les intégrer dans la représentation CORDIS-ANIMA en leur associant des modules spécifiques évoquant leur présence dans la constitution de l'objet global. La description de ces modules comportera les points de communication qui les caractérisent et, quand ce sera possible, la référence à une modélisation de leur structure et de leurs paramètres mécaniques.

Un objet sera dit *complet* quand tous ses modules, y compris les modules transducteurs, seront représentés. Lorsque l'on voudra parler de la partie purement numérique, on dira *l'objet interne*.

#### 4 - Limites de modularité. Les "atomes" de CORDIS-ANIMA

##### *Le plus petit objet...*

Le plus petit objet envisageable est celui qui ne comporterait qu'un seul point de communication. Il y en aurait de deux sortes, selon que ce point est de type M ou de type L. Un atome M correspondrait à un algorithme calculant une position dans l'espace à partir d'une force, un atome L calculerait une force à partir d'une position.

Un système exclusivement constitué d'atomes M et d'atomes L ne peut toutefois donner lieu, compte tenu des règles de connexion, à une variété d'objets. Une seule figure est possible : une étoile formée d'un atome M, environné d'autant d'atomes L qu'on le souhaite (Figure 5). Pour pouvoir construire une plus grande variété d'objets, il faut des atomes un peu plus complexes.

Partant des points isolés, fabriquons des atomes plus complexes en les attachant. Suivant un principe d'économie, cherchons le nombre minimal d'attaches qui permette de constituer un système d'atomes générique. Une rapide investigation montre qu'une seule attache suffit et que le seul cas possible est celui où elle relie deux points L. Dans les deux autres cas on ne peut obtenir en effet que des figures limitées : une étoile à noyau double si l'on construit un atome à l'aide de deux points M, un arbre strict (sans boucle) dans le cas où l'on construit un atome à l'aide d'un point M et un point L (Figure 5).

Par contre, la base constituée d'un atome à deux points L et d'un atome à un point M permet d'obtenir toutes les figures possibles sous les espèces du réseau topologique.

##### *Le point matériel et l'élément de liaison*

Ainsi, la plus petite décomposition possible donnant lieu à un système générique est celle où l'on a deux types d'atomes distincts, l'un correspondant à un point M seul, l'autre à une paire de points L indissociables.

L'atome à un point M a une localisation ponctuelle dans l'espace puisque son algorithme calcule des positions au cours du temps à partir de forces appliquées par son environnement. Il constitue alors la "granulation" de la matière. Nous l'appellons *Point Matériel*.

Si l'on veut respecter pour cet atome, la clause fondamentale selon laquelle tous les objets (composites ou élémentaires) doivent pouvoir être plongés dans le domaine d'expérimentation sensori-motrice, il faut que l'espace simulé puisse s'inscrire dans l'espace réel. Les positions du point matériel sont alors, dans le cas le plus général, des positions dans un espace tri-dimensionnel, ce qui n'empêche pas de considérer des espaces plus simples comme le plan ou la ligne.

L'atome à deux points L n'a de localisation dans l'espace que celle des points matériels auxquels il est nécessairement connecté. Calculant deux forces à destination de ces points à partir des seules choses qu'il connaisse d'eux : leurs positions (et leur variations), il établit une liaison entre eux. Nous l'appelons *Élément de Liaison*.

En raison toujours de la même clause, il faut respecter un principe fondamental en mécanique : celui de l'action et de la réaction. Ainsi, dès ce niveau atomique, faut-il garantir que les deux forces calculées par l'élément de liaison à destination des points matériels auxquels il est connecté soient coaxiales, égales en intensité et de sens opposés.



Le *point matériel* et l'*élément de liaison* sont les constituants de base de l'univers CORDIS-ANIMA (Figure 6).

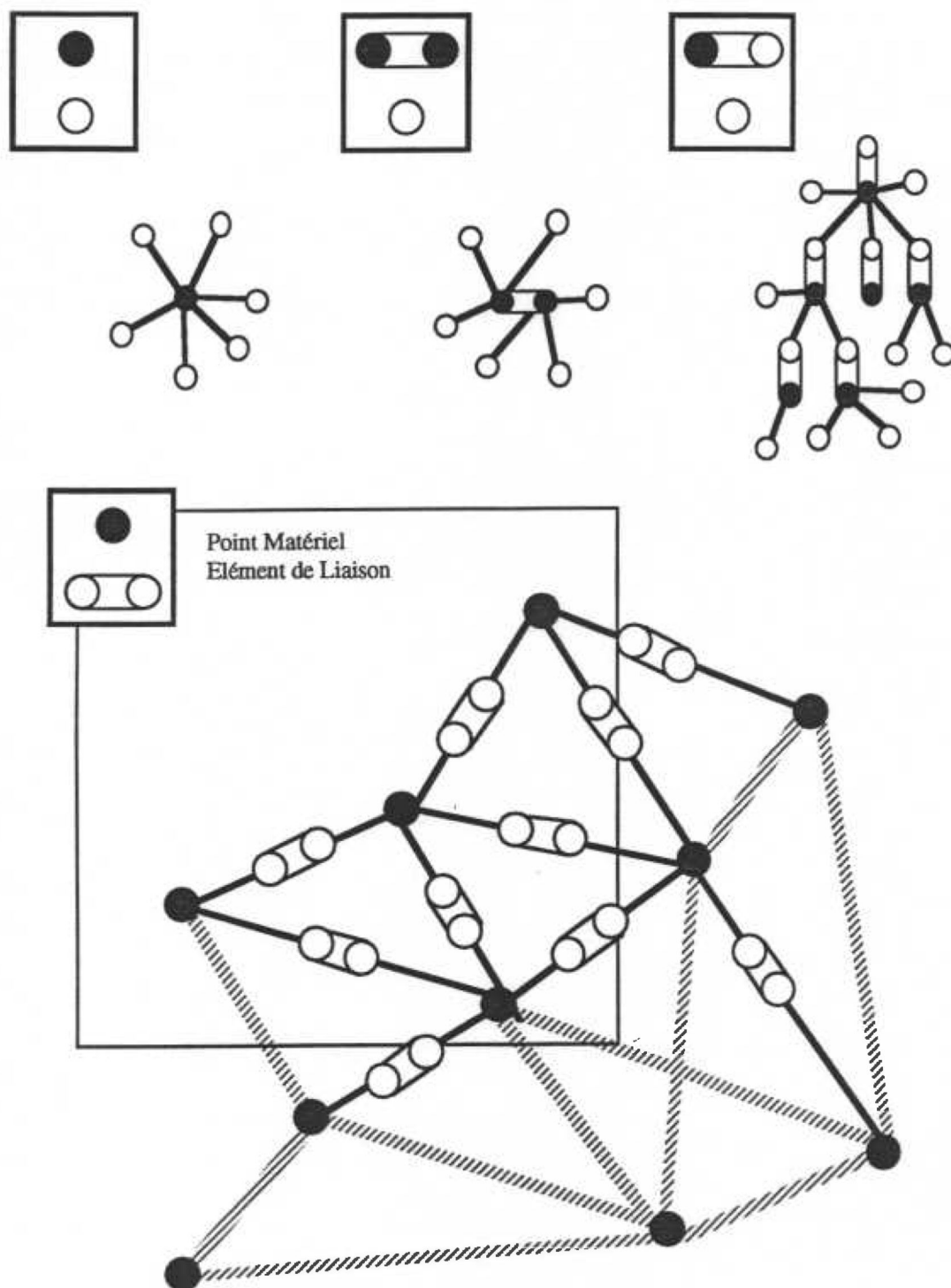


Figure 5

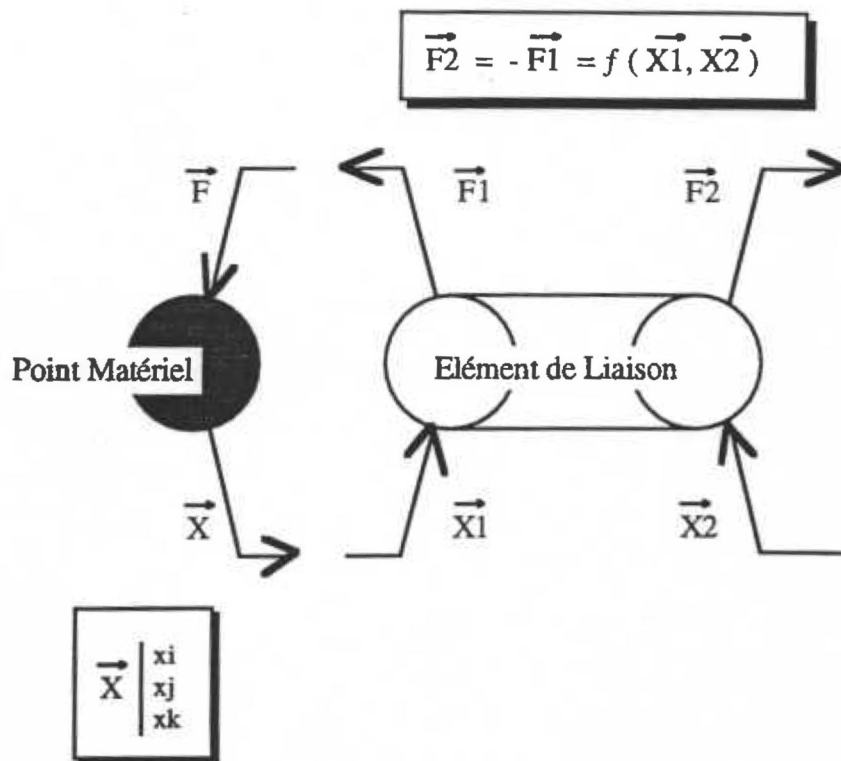


Figure 6

**Remarques :**

- Dans l'expression décrivant la fonction de l'Elément de Liaison, dans la figure ci-dessus, nous faisons figurer une fonction des vecteurs position  $X_1$  et  $X_2$ . Il s'agit d'une expression condensée pour signifier la dépendance de la force calculée par rapport à ce qui se présente aux entrées. Dans le cas le plus général, il faut considérer que l'expression effective fait intervenir les vecteurs position, les vecteurs vitesse, etc. tous calculés cependant à partir des positions et de leurs évolutions.
- Il faut éviter la confusion entre ce que l'on a appelé au début *connexion* et ce que l'on appelle maintenant *liaison*. La *connexion* était (et demeure) la connexion de deux points de communication, prenant la sortie de l'un pour entrée de l'autre et réciproquement. La *liaison* est une liaison mécanique, physique, entre deux points matériels. Elle met d'ailleurs en jeu deux connexions.
- La nature même du support technologique nous a imposé une ponctualisation des communications. Entre l'opérateur et l'objet global tout d'abord, entre les composants de l'objet ensuite. C'est cette contrainte qui, poussée jusqu'au bout, nous conduit à la seule physique possible en simulation numérique modulaire : la physique du point. Dans cet univers, la matière, comme le temps, est nécessairement discrétisée et localisée en des points, et comme pour le temps, il faudra jouer avec les mailles de notre propre système de perception pour que, ne trahissant pas le discontinu, elles nous laissent croire au continu.

- Remarquons enfin que jusque là rien n'a été déterminé quant aux algorithmes effectifs et que tout ce premier cadre formel n'est déduit que des deux contraintes évoquées au début : celle de la discrétisation et finitude des communications et celle de la modularité et de la "physicalité" des modules.

## II - LES MODULES FONCTIONNELS

Nous avons dans le premier chapitre introduit et utilisé le concept de *module physique*. Les entrées et les sorties d'un module physique sont des forces et des déplacements. Il élabore les sorties à partir des entrées selon un calcul spécifique qui prend modèle sur un processus physique élémentaire.

Il est clair, avant même que l'on évoque les algorithmes effectifs de simulation, que cette logique physique ne peut être adoptée sans limites. Les capacités de calcul devraient être infinies. En fait, tout système de représentation doit à un moment ou un autre articuler des niveaux de cohérences disjoints. La pertinence tient alors autant à la façon dont les compromis sont rendus créatifs qu'à la vérité des parties cohérentes.

Ici, le dualisme s'établit entre le physique et le *fonctionnel*.

Ce que nous entendons par modules fonctionnels est une catégorie de modules qui ne manipulent pas exclusivement des variables physiques, ne répondent pas nécessairement à un modèle physique immédiat. Utilisés cependant avec des règles précises aux côtés des modules physiques ils vont réaliser rationnellement certains "racourcis" et permettre, moyennant quelques précautions, d'étendre considérablement les possibilités de représentation.

Il est bien évident que ces modules doivent être introduits avec parcimonie : un système ne peut vivre sans un certain lot d'exceptions mais il ne faut pas que l'exception devienne la règle sans quoi on ne parlera plus de système.

Nous présentons ci-après principalement trois types de modules fonctionnels. Les deux premiers sont voisins (et complémentaires) dans leur fonction. Le troisième correspond à une autre catégorie d'extension.

D'autres fonctionnalités sont envisageables. CORDIS-ANIMA est fondé sur une base stable mais c'est un système "vivant", c'est-à-dire en extension à partir de cette base. Des fonctions autres que celles qui sont présentées ici ont déjà été expérimentées. Dans la mesure où leur systématisation n'est pas achevée, nous nous interdisons toutefois de les présenter pour l'instant.

### 1 - Modules de variation dynamique

Le système est fondé sur la notion d'objet physique. C'est à cette notion qu'est attachée l'idée de *permanence*. C'est elle aussi qui permet de définir des classes de *variation*. En d'autres termes les choix que nous décrivons ci-après caractérisent la dialectique permanence/variation propre au système CORDIS-ANIMA.

Remarquons bien qu'il s'agit là de la variation structurelle *dynamique*, c'est-à-dire celle qui se fait pendant le temps même de la simulation. Soulignons également que cette variation structurelle dynamique ne peut être utilisée sans précautions. En effet, dans la mesure où un système logique se superpose à un système physique, il peut en résulter une certaine incohérence. En particulier, lors de l'établissement ou de la rupture d'une connexion, une discontinuité d'états physiques des constituants en cause peut provoquer l'équivalent d'un apport ou d'une suppression d'énergie.

### *Modules de variation paramétrique dynamique*

On peut introduire une notion semblable pour la variation paramétrique dynamique. Dans le même esprit, on définit alors un type de modules fonctionnels permettant à l'opérateur ou à un sous-objet, d'agir sur les paramètres d'un algorithme.

Les *modules de variation paramétrique dynamique* se présentent donc, dans leur forme générale, avec un certain nombre de points M ou L qui les rendent connectables aux points L ou M des objets engendrant le contrôle. Ils possèdent, comme le premier type de modules de contrôle structurel, des sorties non homogènes aux points M et L, portant les variations des paramètres en question. Ces sorties sont connectables à des entrées d'un type que nous n'avons pas encore évoquées jusque là : les *entrées de paramètres*. Les entrées de paramètres sont les lignes par lesquelles on accède, pendant le temps même de la simulation, à un sous-ensemble des paramètres d'un module : les paramètres dits *contrôlables*.

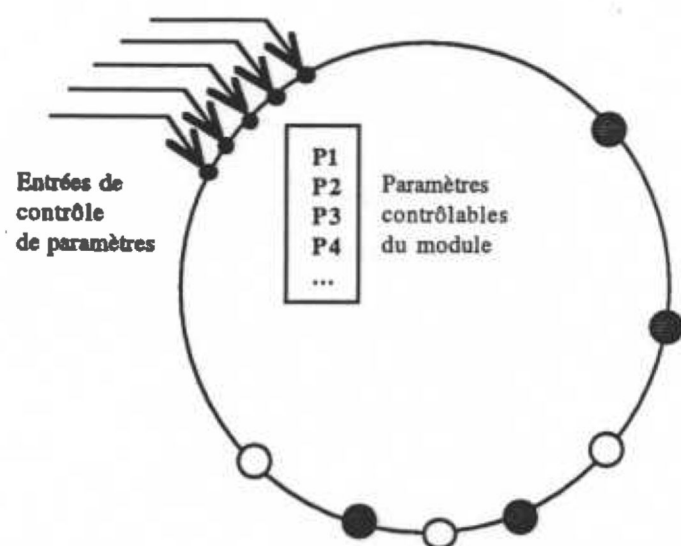
En fait, tout module est a priori doté de paramètres, mais, pour des raisons relevant parfois de la technique, parfois de la signification de la modélisation, le contrôle dynamique ne se justifie pas pour tous les paramètres. D'où la notion de *paramètres contrôlables dynamiquement* et, par opposition, de *paramètres non contrôlables dynamiquement*. Les derniers sont bien évidemment contrôlables également, mais *statiquement*, ce qui signifie essentiellement en dehors du temps de simulation.

Notons alors que l'on doit compléter la représentation formelle générale des modules, physiques ou fonctionnels en leur attachant une liste de paramètres composée des paramètres contrôlables et des paramètres non contrôlables.

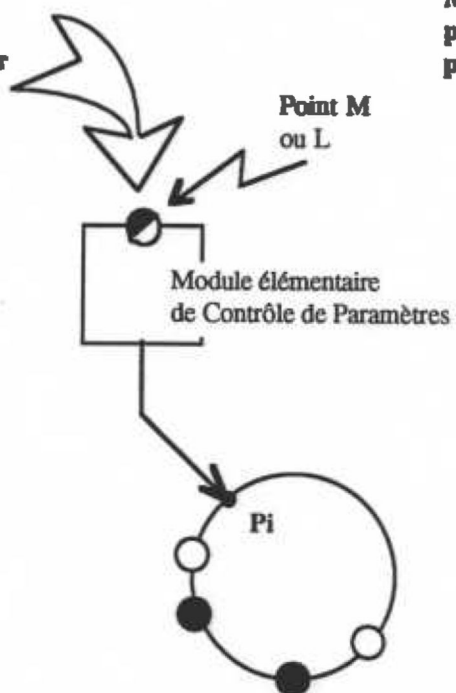
Le module de contrôle de paramètres le plus simple présente un point M ou L et une sortie de contrôle (Figure 9).

#### *Remarque :*

En tant que module, le module de contrôle de paramètres comporte lui aussi des paramètres. Toutefois, le contrôle des paramètres d'un module de contrôle de paramètres ne présente aucune espèce d'intérêt particulier.



Modification  
paramétrique  
par l'opérateur



Modification  
paramétrique  
par contrôle interne

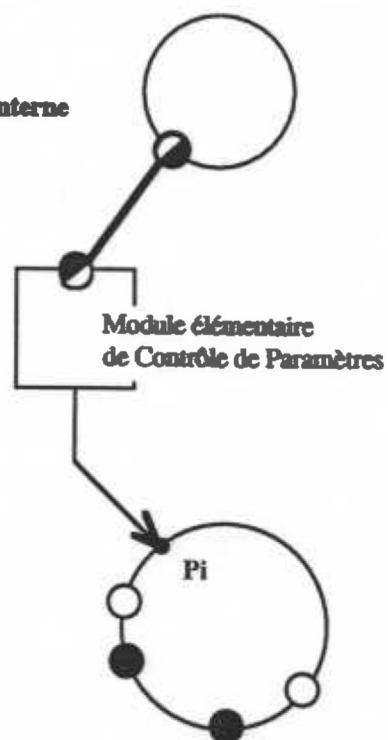


Figure 9

## 2 - Modules de changement de point de vue relatif

Nous allons maintenant introduire un troisième type de modules fonctionnels : les modules de changement de point de vue relatif.

Revenons à l'objet global, connu seulement par la liste de ses points de communication. Nous disposons, pour chacun de ces points, des représentations informatiques des variables forces ou positions. Rien ne nous interdit, avant de les associer à leur transducteurs respectifs, de les amplifier ou de les réduire arbitrairement. Plus exactement, entre la représentation numérique interne de ces variables et les phénomènes physiques réels auxquels elles correspondent il y a une relation d'échelle que l'on doit établir explicitement.

Ainsi, lorsque nous avons affaire à une certaine valeur " $\phi$ " en unités internes de la variable F, il faut que lui corresponde une valeur "f" en newtons, soit en unités de force du système standard (MKS). De même, pour une valeur " $\xi$ " en unités internes de la variable X, il faudra que lui corresponde une valeur "x" en mètres. Plus généralement, entre les unités fondamentales d'un système et celles de l'autre, la correspondance doit être établie de manière explicite, ce qui signifie qu'entre  $\phi$  et f,  $\xi$  et x, le temps réel (t) et sa représentation en machine, respectivement, existent des coefficients de correspondance.

Supposons les alors désignés et définis à l'aide des relations suivantes :

$$\begin{aligned}f \text{ (en newtons)} &= R_f \cdot \phi \text{ (en unités de forces internes)} \\x \text{ (en mètres)} &= R_x \cdot \xi \text{ (en unités de longueur internes)} \\t \text{ (en secondes)} &= T_e \cdot \tau \text{ (en unités de temps internes)}\end{aligned}$$

Alors, étant donné un objet simulé, si l'on change l'un quelconque ou plusieurs de ces coefficients sans changer le processus de simulation, il se produit globalement pour l'observateur un changement de "taille" de l'objet. On peut alors, de ce qui est une nécessité au départ, faire une latitude en soi et introduire un concept explicite de *coefficient de représentation*. Le coefficient de représentation est défini par la donnée des trois coefficients précédents et permet, étant donné un objet, de décider de sa taille effective dans l'espace de l'observateur.

Dans la mesure où ce qui apparaît là est essentiellement une question de relation entre l'observateur et l'objet, on peut tout aussi bien imaginer, à l'inverse de l'interprétation précédente, que l'objet reste de taille fixe, et que l'observateur a la faculté de changer de point de vue, plus exactement de taille lui-même, ce qui est sans doute plus grisant...

Ce serait dommage de s'en arrêter là. Nous avons constaté à plusieurs reprises qu'en de nombreux points, la relation entre l'observateur et l'objet est de la même nature que la relation entre les sous-objets eux-mêmes. Aussi pouvons nous considérer le même principe de point de vue relatif entre deux sous-objets internes à l'objet, et même le généraliser. Voici alors le cadre de ce troisième type de modules fonctionnels de CORDIS-ANIMA, modules que nous appellerons *modules de point de vue relatif* (dans la pratique nous dirons "MPVR" pour raccourcir).



Extérieurement, le MPVR ne se distingue pas des modules physiques dans la mesure où il ne présente que des points de communication de type M ou L. Il n'est cependant pas un module physique car il ne simule le comportement d'aucun objet physique matériel ou identifiable. Il s'interpose entre deux sous-objets préalablement définis (l'un des deux pouvant être d'ailleurs l'opérateur lui-même) et, pourrait-on dire, il "convertit" un certain nombre de points M (respectivement L) en un certain nombre de points L (respectivement M). L'exemple le plus simple (Figure 10) est celui où, rompant une connexion entre un point M et un point L, on introduit à la sortie du premier un facteur multiplicatif "p" (ou un facteur multiplicatif "1/p" à son entrée, ce qui revient au même). Dans ce cas, au travers de ce point de connexion, les objets se "voient" mutuellement selon une taille relative différente.

(Note : nous employons le mot "taille" comme métaphore, sa signification précise suppose une connaissance plus approfondie des propriétés des objets).

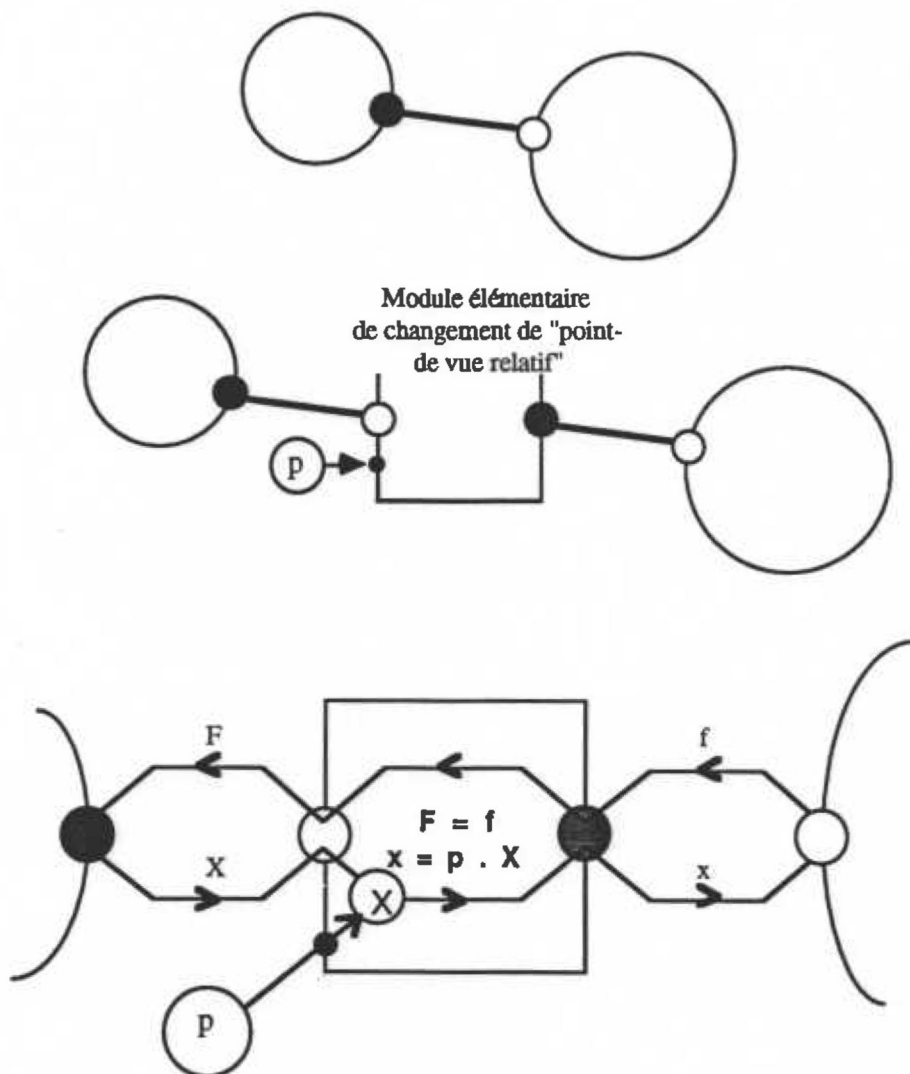
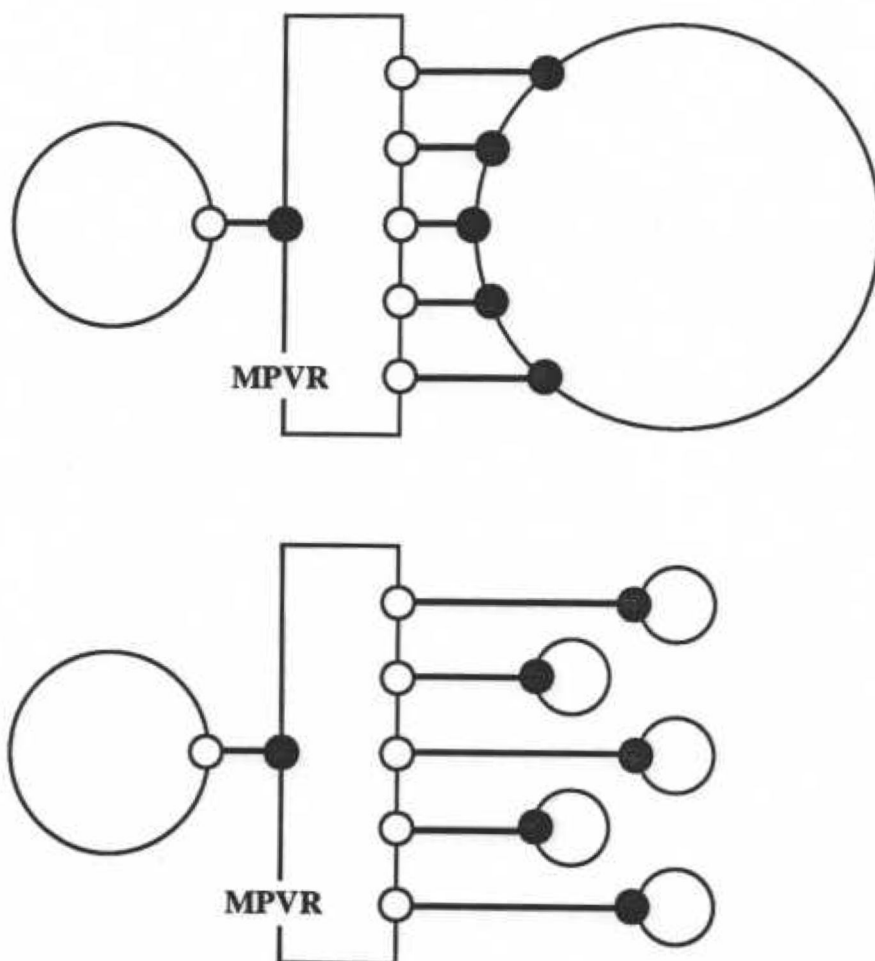


Figure 10

Un cas plus complexe peut correspondre à un changement de variables permettant d'appliquer par exemple une logique physique facilement concevable dans un repère cartésien à un repère en coordonnées polaires. Dans ce cas, le nombre de points d'un côté et de l'autre est le même.

Enfin, dans un certain nombre d'applications fondamentales, il est tout-à-fait important de pouvoir réduire le nombre de "point d'accès" à un objet et, par exemple de distribuer les influences d'un petit nombre de points L (ou M) d'un premier objet, sur un grand nombre de points M (ou L) d'un second objet, ou encore sur un grand nombre de points M (ou L) appartenant à plusieurs objets distincts.

Dans le cas général la simple multiplication du cas élémentaire fait place à un traitement plus complexe. L'exemple le plus typique est celui de la *synthèse modale*. Nous y reviendrons plus loin.



**Figure 11**  
Module de Changement de Point de vue Relatif (MPVR)

### 3 - Conclusion de la première partie

Dans toute cette première partie, nous avons donc introduit le cadre général de CORDIS-ANIMA, son enveloppe formelle, déduite exclusivement de quatre objectifs initiaux :

- 1) Simuler un objet physique accessible à l'expérience sensori-motrice multi-sensorielle.
- 2) Le faire à l'aide d'une machine numérique pourvue des transducteurs électro-physiques associés aux trois canaux sensoriels principaux (geste, ouïe, vue).
- 3) Proposer une dualité *permanence/variation* où la permanence est attachée à la notion d'objet physique, la variation à la transgression de cette notion selon deux modes : la *variation structurelle* et la *variation paramétrique*.
- 4) Disposer enfin, pour cette démarche, d'un langage.

Ayant donné cette forme, il nous faut maintenant donner la "matière", c'est-à-dire pénétrer à l'intérieur des modules physiques pour en décrire les algorithmes.

## DEUXIEME PARTIE

### Les algorithmes de CORDIS-ANIMA

Nous avons jusque là fait abstraction de l'intérieur de la machine, nous n'avons considéré que la nécessité de représenter les phénomènes et échanges physiques sous forme de signaux uni- ou multi-dimensionnels.

Nous allons maintenant plus loin en tenant compte de deux ordres de détermination supplémentaires : celui de la réalité interne des machines et celui de la réalité du monde physique que nous voulons leur faire représenter, en prenant des objets réels comme références pour les modules physiques.

## I - LA MACHINE

### 1 - Représentations numériques, opérateurs, etc

Ce n'est pas le lieu ici de décrire les machines numériques, la tâche serait sans fin d'ailleurs puisque elles sont en évolution permanente. On peut toutefois inventorier rapidement les quelques matériaux auxquels on aura nécessairement affaire.

Tout d'abord, une machine numérique, quelle que soit son architecture, qu'elle soit séquentielle, parallèle, massivement parallèle, n'utilise encore aujourd'hui rien d'autre que les éléments suivants : les bits d'information, matérialisés de plusieurs façons selon qu'ils sont mémorisés ou transmis, la mémoire, la transmission, le traitement de ces bits d'information à l'aide des opérateurs logiques booléens et des tables de vérité, le programme et... une horloge.

Le niveau dont nous avons besoin ici est celui où les bits s'organisent en *mots binaires* représentant des nombres entiers ou réels représentant à leur tour les paramètres et variables physiques. Celui également où les opérateurs booléens se structurent en opérateurs plus complexes réalisant les comparaisons, additions, soustractions, multiplications, divisions, etc. sur ces représentations numériques. Les programmes sont des séquences d'instructions effectuant sur les types d'opérandes évoqués les opérations ci-dessus. A signaler l'usage particulier que nous aurons entre autre du déroutement conditionnel, c'est-à-dire la possibilité de changer le cours des opérations en fonction du résultat des opérations précédentes.

Nous ne donnerons pas plus de précision ici et ne ferons en particulier aucune hypothèse restrictive quant à la nature séquentielle ou parallèle des processus ou au nombre de machines (de processeurs) à utiliser.

Nous allons par contre évoquer maintenant de nouvelles contraintes au problème de la simulation physique par une machine numérique.

## **2 - Discrétisation temporelle, finitude du temps de calcul** ***T-Simulabilité d'un objet physique***

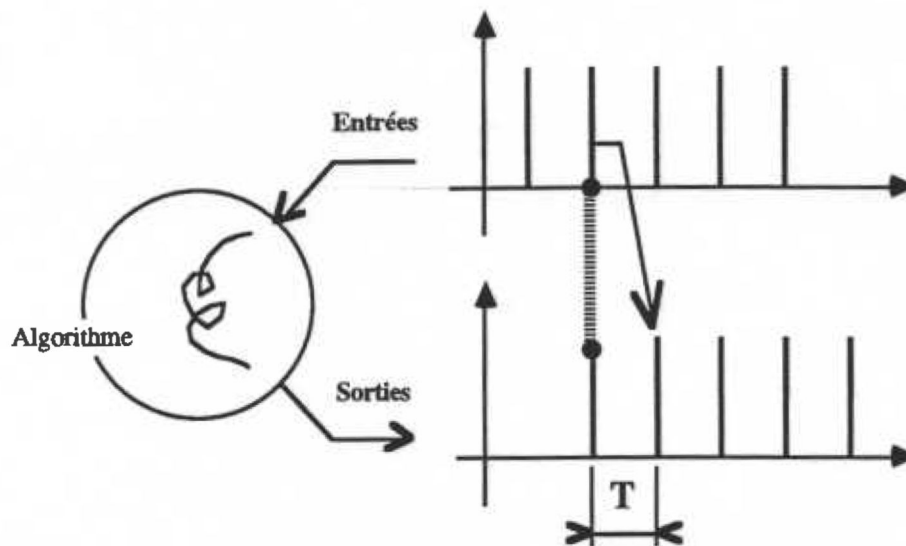
Les grandeurs physiques (variables ou paramètres) sont représentées par des mots binaires organisés suivant ce que l'on appelle en abrégant des **ENTIERS** (à ne pas identifier avec les entiers naturels) ou des **FLOTTANTS** etc. Il s'agit là d'un niveau de discrétisation supplémentaire puisque dans un cas comme dans l'autre on utilise un nombre fini de bits. Mais il y a une troisième discrétisation indissolublement liée aux précédentes : quelle que soit la façon d'associer une représentation numérique à un état ou de traiter les états pour les faire varier, il existe une granulation du traitement, un laps de temps minimal, en général géré et homogénéisé par une horloge fondamentale unique, pendant lequel la variable ou la caractérisation de l'état restent fixes.

Vu de l'extérieur, cette discrétisation temporelle a deux conséquences. D'une part, les signaux sont nécessairement des suites numériques, plus précisément des suites de mots binaires **ENTIERS** ou **FLOTTANTS**, on dira des suites d'échantillons ou *signaux échantillonnés*. D'autre part, entre les échantillons correspondant aux entrées et les échantillons correspondant aux sorties résultant d'un calcul sur ces entrées, il y aura un temps d'élaboration incompressible. Aucune machine numérique n'est capable aujourd'hui de traiter instantanément, ne serait-ce que pour la transmettre, une donnée.

Nous adoptons alors, pour des raisons de généralité de représentation, un schéma temporel spécifique et unique pour les entrées-sorties et nous appelons *T-Simulation* la situation dans laquelle un algorithme de simulation :

- *traite en entrée des signaux échantillonnés aux conditions de Shannon à la période  $T$  (constante),*

- produit en sortie également des signaux échantillonnés selon ces conditions,
- et établit entre les entrées et les sorties la relation spécifique suivante : les instants d'entrée et les instants de sortie sont synchrones, et entre les entrées et les sorties qui en résultent, il s'écoule une seule période  $T$ . (Figure ci-dessous).



**Figure 12**  
Discrétisation temporelle et T-Simulabilité

Nous qualifions alors de *T-Simulables* les objets physiques dont on peut donner une simulation juste, selon des critères de précision d'observation prédéfinis, dans ces conditions.

$T$  est la période d'échantillonnage des signaux d'entrée et des signaux de sortie, c'est également ce que nous appellerons la *période de simulation*. Pour  $F$ , l'inverse de  $T$  ( $F = 1/T$ ), nous parlerons de fréquence d'échantillonnage quand il s'agira des signaux, de fréquence de simulation quand il s'agira de la boucle de calcul du module.

Cette notion de T-Simulabilité a un certain nombre d'incidences importantes sur les conditions d'implantation des modules de simulation. Nous n'entrerons pas dans le détail des questions qu'elle soulève dans le cadre de cet article, nous nous contenterons des quelques considérations suivantes :

- Tout d'abord, un objet peut ne pas être simulable pour une valeur  $T$  donnée, mais l'être pour une valeur  $T'$  plus petite. Considérons alors la plus grande valeur  $T_e$  de la période de simulation pour laquelle un objet est simulable. On parlera de *bande passante comportementale* ou interne à propos de la grandeur  $F_m = 1/2T_e$ . Cette donnée signifie que, quelle que soit la variable à l'aide de laquelle on observe le comportement interne de l'objet, sa fréquence d'évolution n'excède pas  $F_m$ .

Si l'on s'avise alors, faute de moyens, de simuler à la période  $T$  un objet non  $T$ -Simulable, mais  $T_e$ -Simulable, avec  $T_e < T$ , il s'ensuit une dérive dont la gravité dépendra du rapport entre  $T_e$  et  $T$ . Cette dérive est comparable à la conséquence du sous-échantillonnage dans le traitement des signaux.

- Des objets, par contre, représentés par certains modèles idéalisés, ne sont pas  $T$ -Simulables quel que soit  $T$  aussi petit que l'on veut. Prenons par exemple le cas d'un ressort idéal (sans inertie) que l'on caractériserait par la relation suivante :

$$F(t) = K.\Delta X(t)$$

où  $F$  est la force qu'il oppose à une déformation  $\Delta X$ , et  $K$  sa constante de raideur. et  $F$  la sortie, il faut que l'égalité soit instantanée. Un tel objet n'est pas simulable, et ne sera jamais simulable par aucune machine numérique aussi rapide soit-elle.

Nous arrêtant là de ces considérations, nous signalerons simplement qu'il est possible, dans un contexte global où sont en cause plusieurs composants d'"emprunter" aux objets  $T$ -Simulables le temps de calcul qu'ils ont éventuellement en excédent pour le donner aux objets du type que l'on vient de voir de telle sorte que l'on puisse "simuler" l'égalité instantanée. Une des conditions nécessaires (non suffisante) est cependant que ces objets n'aient aucun point de communication avec l'extérieur.

Ceci a deux conséquences :

- Il y a là une contravention à la clause d'expérimentabilité des objets élémentaires puisque ces objets spéciaux ne peuvent être simulés (donc expérimentés) isolément ;
- par ailleurs, cette donnée imposera une contrainte particulière au niveau des protocoles de communication entre les processus réalisant, sur le ou les processeurs en cause, les modules de simulation concernés.

Nous supposons toutes ces questions résolues pour aborder la suite de notre développement.

## II - LES ALGORITHMES DE BASE

### 1 - La "matière linéaire"

Le système des algorithmes, à l'inverse du cadre formel, est élaboré en partant des atomes. Le pari est ici de tout construire à partir de ceux-ci : du *point matériel* et de l'*élément de liaison*, sachant toutefois qu'ils peuvent présenter diverses variétés ; et la démarche consiste à chercher le meilleur principe d'économie à chaque niveau de cette construction.

Pour élaborer l'algorithme d'un composant physique, on peut adopter une première attitude : partir des équations mécaniques de cet objet, associer aux variables et paramètres des



grandeurs numériques, aux opérateurs les opérateurs numériques correspondants. L'algorithme est en quelques sortes la vérification de l'équation ainsi transposée à temps-espace discrets et variables quantifiées, pour toutes les valeurs (discrètes) consécutives du temps. Mais une attitude plus judicieuse consiste à partir de la machine : chercher, en mettant le minimum d'éléments algorithmiques en œuvre, le premier processus dont le comportement permette de l'identifier de manière acceptable à l'objet de référence.

Les deux démarches vont naturellement à la rencontre l'une de l'autre, mais dans la première il se peut que l'on s'encombre de propriétés qui n'ont pas d'intérêt pertinent par rapport à l'objectif visé. Par ailleurs la seconde pose plus explicitement le problème de la représentation des objets à l'aide des processus numériques.

Dans la genèse des algorithmes de CORDIS-ANIMA, il faut le signaler, nous avons procédé de la seconde manière, tournant délibérément le dos au départ aux théories du filtrage, aux équations différentielles, ... même si celles-ci sont réapparues très vite, pour ne raisonner qu'en termes d'économie algorithmique.

Ainsi, le paradigme des suites récurrentes s'est-il imposé naturellement, de la manière suivante :

Nous cherchons à réaliser un processus dont la structure est permanente, capable d'évoluer de lui-même mais aussi sous l'influence d'un phénomène extrinsèque.

Dans le contexte numérique qui nous est donné, évolution signifie suite d'états et l'état, dans la vision la plus simple, c'est une grandeur numérique (un mot binaire).

Le problème est alors le suivant :

Etant donné l'état d'un objet représenté par une variable numérique  $X(n)$  à l'instant  $n$ , étant donné une intervention extérieure représentée elle aussi par une variable numérique  $Y(n)$  (a priori susceptible de changer à chaque instant), étant donné encore que nous disposons de la possibilité de mémoriser ces variables, que nous disposons d'opérateurs (les "additions", les "multiplications" numériques...), comment organiser le tout pour engendrer un comportement dynamique utile et maîtrisable ?

La première application du principe d'économie consiste à dire que pour chaque nouvel instant on va essayer de mettre en œuvre le minimum de valeurs et le minimum d'opérateurs ; d'où l'idée d'utiliser comme valeurs les états passés, d'en utiliser le moins possible et de prendre le passé le plus proche.

On s'aperçoit alors assez vite que ce matériau n'est pas tout à fait suffisant pour engendrer des phénomènes variés et qu'il faut recourir à d'autres types de valeurs que les états et les interventions extérieures. La notion de paramètres arrive strictement ici, pour cette fonction et dans cette définition : ce sont les valeurs intervenant dans le processus, qui ne sont ni des états ni des variables extérieures et qui sont nécessaires à la constitution d'un processus dynamique. A côté de la définition structurelle qualitative du processus, ils lui confèrent sa permanence quantitative.

Après quelques investigations, un déterminisme intrinsèque nous conduit vers une forme naturelle et générale du processus dynamique numérique correspondant à ce paradigme du calcul récurrent qui peut s'écrire de la manière suivante :

$$X(n) := A.X(n-1) + B.X(n-2) + \dots + U.Y(n) + V.Y(n-1) + W.Y(n-2) + \dots + P$$

A, B, C, ... U, V, W, ... P sont des paramètres.

On appelle en général cela un "filtre numérique". La théorie des filtres numériques est bien connue, elle nous est fort utile dans CORDIS-ANIMA et dans tout ce propos de modélisation. Toutefois nous ne pouvons nous résoudre à appeler cela simplement "filtre" parce l'on ne désigne par là qu'une conséquence, une propriété. Le filtre est une boîte noire, on observe ce qui entre et ce qui sort. Nous voulons pointer ce qui est dans la boîte, c'est la que nous cherchons la substance de l'objet.

Remarquons au passage, que lorsque U n'est pas nul, on tombe sur une forme de représentation typique d'un objet non T-Simulable quel que soit T aussi petit que l'on veut, puisqu'apparaît l'indice temporel "n" dans les deux membres de la relation.

Ainsi, cette forme de calcul, linéaire, nous sert-elle de base pour une première catégorie de composants.

Il s'agit maintenant d'adopter des cas précis et de donner une signification aux paramètres. Toujours selon le même souci de simplicité et d'économie, on cherche naturellement à mettre en œuvre le processus précédent à l'ordre le plus petit. L'ordre, c'est-à-dire le nombre d'itérations pendant lesquelles on devra mémoriser les états.

Ainsi, l'ordre 0 (aucune mémoire) nous donne-t-il :

$$X(n) := U.Y(n) + P$$

L'ordre 1 :

$$X(n) := A.X(n-1) + U.Y(n) + V.Y(n-1) + P$$

L'ordre 2 :

$$X(n) := A.X(n-1) + B.X(n-2) + U.Y(n) + V.Y(n-1) + W.Y(n-2) + P$$

Ces trois formules nous permettent d'englober les comportements mécaniques fondamentaux comme le point fixe ( $U = 0$ ), le ressort ( $U \neq 0$ ) avec l'ordre 0, le frottement avec l'ordre 1, l'inertie, l'oscillateur harmonique, avec l'ordre 2. Il faut toutefois organiser ces expressions et donner une signification aux paramètres.

De multiples variantes sont possibles. C'est alors qu'un autre ensemble de considérations complète ces considérations algorithmiques : Afin de simplifier la représentation (mentale), on déclare que toute la matière inerte (possédant une inertie) est concentrée dans les points matériels, et que les éléments de liaisons régissent exclusivement les interactions entre ces éléments ponctuels de matière. Les points matériels deviennent alors des *masses ponctuelles* et les liaisons des *interactions mécaniques* idéalisées.

Soulignons que ce choix simplifie les représentations mais nous place d'emblée sous le coup de la non simulabilité pour les interactions mécaniques, corrolaire d'ailleurs, de leur idéalisation.

Ainsi, les trois éléments fondamentaux de la matière de CORDIS-ANIMA sont la *Masse*, le *Ressort*, le *Frottement* [9], [15].

L'algorithme de la masse est obtenu en identifiant la formule d'ordre 2 précédente à l'équation différentielle de la loi fondamentale de la dynamique :

$$F = M.\Gamma$$

Cette identification peut se faire de plusieurs façons selon le type d'approximation que l'on utilise pour la représentation discrète de la vitesse et de l'accélération. Nous donnons la plus simple :

$$\begin{aligned} V(n) &= X(n) - X(n-1) \\ G(n) &= V(n) - V(n-1) = X(n) - 2.X(n-1) + X(n-2) \end{aligned}$$

Dans cette représentation, l'écart de temps  $\Delta t$  entre deux échantillons est considéré comme égal à 1, ce qui simplifie notablement les expressions. Mais du même coup, cela fixe implicitement le coefficient de représentation temporelle  $T_e$  que nous avons introduit dans la première partie (au § 2.2.) et qui n'est autre, alors, que la valeur de la période d'échantillonnage.

Ainsi, obtient-on pour la masse ponctuelle l'algorithme suivant :

$$X(n) := 2 \times (n-1) - X(n-2) + (1/M) F(n-1)$$

$X(n)$  : position calculée,  
 $X(n-1)$  : mémoire de la position précédente,  
 $X(n-2)$  : mémoire de la position deux instants précédents,  
 $F(n-1)$  : force d'entrée,  
 $M$  : valeur de la masse.

Les variables  $X$  et  $F$  peuvent être considérées comme des variables uni-dimensionnelles ou comme des vecteurs à deux ou trois composantes, ce qui ne change rien à la forme de l'algorithme.

Le ressort et le frottement ont donné lieu à des algorithmes répondant de la même forme générale dans les premières versions de CORDIS [8], [15], ne considérant que des liaisons unidimensionnelles entre les masses, ce qui, pour un nombre intéressant de situations, est suffisant. Les algorithmes sont de la forme :

$F2(n) = -F1(n) = -K.(X2(n)-X1(n))$  pour le ressort

$F2(n) = -F1(n) = -Z.(V2(n)-V1(n))$  pour le frottement.

X1 et X2 sont les positions des masses dans l'espace uni-dimensionnel, V1 et V2 leurs vitesses.

K et Z sont respectivement la constante de raideur du Ressort et la constante de frottement du Frottement.

On peut remarquer au passage, donc, que ces modèles ne sont pas T-simulables, puisque de part et d'autre du signe "=" figurent des variables avec l'indice temporel "n".

Simulé individuellement, le "Ressort" ainsi défini a comme algorithme :

$$F2(n) = -F1(n) : = -K.(X2(n-1)-X1(n-1))$$

où la force obtenue en sortie a un retard de  $T_e$  par rapport à la valeur qu'elle devrait avoir. Il en serait de même pour le Frottement.

Cette dérive n'a pas d'interprétation mécanique simple. Il faut noter toutefois que son effet, dans le cas où l'élément est simulé isolément, n'est pas très sensible.

On remarquera également que le ressort ainsi défini a une longueur nulle au repos. En fait, dans ce premier système de représentation, suffisant pour un grand nombre de cas relatifs au son, le réseau est essentiellement topologique : les points matériels se déplacent dans un espace tridimensionnel, mais ils sont tous confondus en une seule et même origine au repos. L'espace de déformation est tridimensionnel, mais l'espace de l'objet global est de dimension nulle.

Dans les versions plus récentes de CORDIS-ANIMA, en raison de la nécessité d'une détermination spatiale des objets animés, les liaisons entre masses sont réellement tri-dimensionnelles, et les ressorts ont une longueur au repos non nulle. Les algorithmes s'écrivent respectivement :

$$F2 = -F1 = - K.((X2-X1) / \|X2-X1\|).\Delta L$$

$$F2 = -F1 = -Z.((X2-X1) / \|X2-X1\|).\Delta V$$

où F1, F2, X1 et X2 sont cette fois des vecteurs, et où  $\Delta L = \|X2-X1\| - LO$  (longueur au repos).

Dans ces expressions interviennent des calculs de distances dans un espace à trois dimensions nécessitant des opérateurs *division* et *racine carrée*.

Le substrat matériel de CORDIS-ANIMA est constitué de ces trois premiers modules que l'on réduit à deux en considérant en général le Ressort et le Frottement comme intégrés en un seul *module de liaison mécanique*.

*Remarque :*

- On a pu constater qu'il y a deux présentations possibles pour les paramètres, se déduisant l'une de l'autre par une expression algébrique. L'une correspond au point de vue du modèle : on parle de la masse (M), de la raideur (K), et du frottement (Z), ce sont les *paramètres physiques* ; l'autre correspond au point de vue algorithmique : ce sont les *paramètres algorithmiques* A,B,C,D etc. de l'équation de récurrence. Lors d'une variation dynamique des paramètres il faudra, à chaque pas temporel, calculer les paramètres algorithmiques correspondant, selon une expression qui, dans certains cas, peut ne pas être simple.

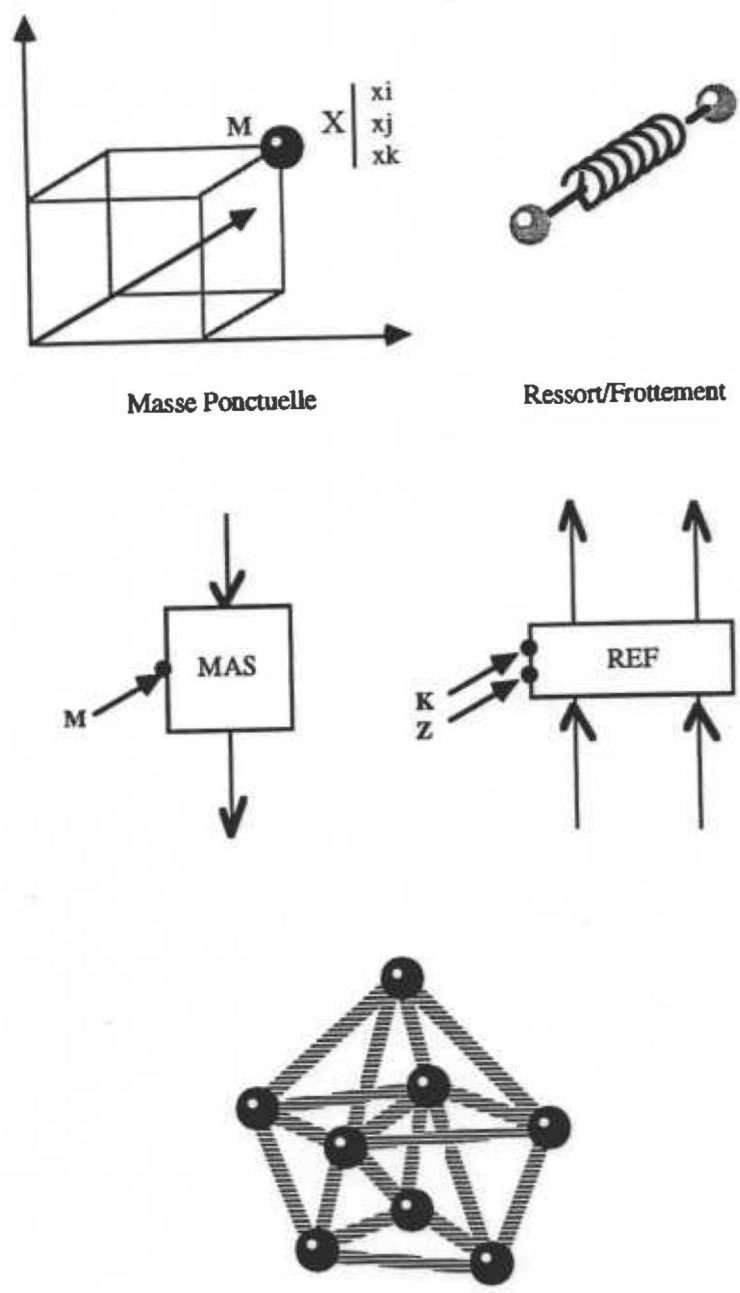
Alors, tablant sur le fait que les variations de paramètres sont, sauf situation particulière, d'un ordre de grandeur inférieur à leur valeur et que, pour de petites variations il y a en général équivalence au premier ordre entre paramètres physiques et paramètres algorithmiques, on peut sans affecter sensiblement la cohérence physique, appliquer les variations directement aux paramètres algorithmiques. Ceci permet une économie appréciable.

On peut maintenant plonger ces deux premiers modules physiques effectifs dans le formalisme général faisant apparaître en particulier les paramètres contrôlables (Figure 13).

Munis de ces éléments nous pouvons maintenant construire un substrat physique. Toutefois, les possibilités sont encore limitées, il faut faire intervenir d'autres propriétés. Tout d'abord, la nécessité des non-linéarités dans l'objet physique : si le dispositif global est rigoureusement linéaire, on ne peut espérer, par exemple, lui faire produire des vibrations sonores à partir d'actions gestuelles puisque celle-ci sont par essence non-audibles. Ensuite, les objets que l'on peut construire à l'aide des éléments précédents sont essentiellement monolithiques, d'un seul tenant, modelable par l'intermédiaire des transducteurs gestuels.

En ce qui concerne la non-linéarité, nous disposons déjà de plusieurs possibilités. Bien que les éléments précédents pris individuellement soient linéaires, ils introduisent un premier type de non-linéarité : celle qui résulte de leur combinaison en structures géométriques tri-dimensionnelles. Par ailleurs, deux autres catégories de non-linéarités résident dans les contrôles structurel et paramétrique dynamiques. À l'aide de ces seules ressources, il n'est pas possible cependant de modéliser et simuler la relation spécifique qui s'établit entre le crin d'un archet et la corde vibrante.

Quant au second point, il est important, aussi bien pour les objets animés visibles que pour les objets manipulés audibles, de pouvoir considérer des "pièces", des "morceaux" distincts susceptibles d'entrer en interaction les uns avec les autres. Pour ces deux dernières catégories de problèmes en particulier, nous avons introduit dès les premières versions de CORDIS la notion de *Liaison Conditionnelle*.



**Figure 13**  
Les modules physiques linéaires



## 2 - La Liaison Conditionnelle

La liaison conditionnelle est une liaison ressort-frottement dont l'état peut changer en fonction de conditions sur ses variables d'entrées. Ce dispositif a déjà été décrit par ailleurs [8], [15], [21], nous en rappelons les principes ici.

La liaison conditionnelle de la première version de CORDIS calcule un module ressort-frottement unidimensionnel selon la formule de base :

$$F2 = - F1 = - K.(X1 - X2) - Z.(V1 - V2)$$

où  $X1$ ,  $X2$ ,  $V1$ ,  $V2$ , sont les positions et les vitesses aux extrémités respectives,  $K$  et  $Z$ , respectivement les constantes de raideur et de frottement.

Le calcul fait cependant intervenir une variable supplémentaire, une longueur au repos fictive, susceptible d'être modifiée en cours de route pour certaines applications. Nous la désignons par la variable  $LA$ . L'élongation est :

$$L = X1 - X2,$$

et la longueur fictive à chaque instant :

$$LR = L - LA.$$

Nous appelons également :  $V$ , la vitesse relative des extrémités à chaque instant.

La formule précédente devient alors :

$$F1 = - K.LR - Z.V$$

Le principe de modification conditionnelle est le suivant :

La modification de la liaison se réalise entre autre par une modification des constantes  $K$  et  $Z$ . Le cas le plus simple étant leur annulation qui permet de "couper" la liaison, mais la forme la plus générale autorise l'affectation d'une valeur particulière à  $K$  et à  $Z$ , parmi un ensemble pré-défini.

### *1 - Etat de la liaison.*

La liaison est alors à chaque instant, caractérisée par un "état", repéré par un indice, et un certain nombre de conditions relatives aux variables d'entrée permettent de changer d'état.

### *2 - Conditions de changement d'état.*

Ces conditions s'expriment de la façon générale suivante :

argument de condition , argument de variable , (paramètre)

Les conditions types sont applicables à toutes les variables, c'est-à-dire à X1, X2, L, LR, V, ce sont :

AUG	:	la variable	augmente,
DIM	:	"	diminue,
CHS	:	"	change de signe,
SUP, param.	:	"	est supérieure à param,
INF, param.	:	"	est inférieure à param.

A ces conditions s'ajoute une "condition extérieure" : CDX, qui, connectée à la sortie d'un module de contrôle structurel permet de commander le changement d'état par l'opérateur ou par une autre partie de l'objet.

A chaque état défini est associé un ensemble de conditions permettant de passer dans un autre état. Ce que l'on peut décrire de la manière suivante :

Dans état (indice d'état) :

si, condition , passer dans l'état indice d'état,  
si, condition , passer dans l'état indice d'état,  
etc ...

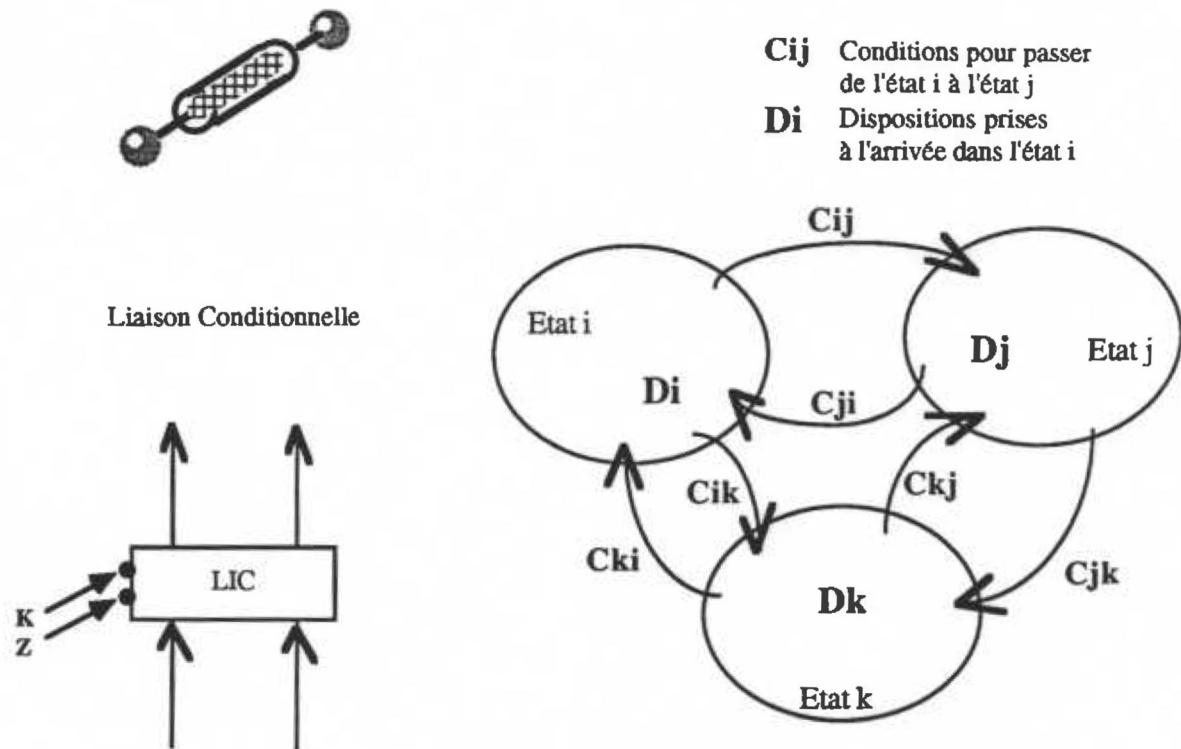
La liste de conditions n'est pas limitée. Plusieurs conditions différentes peuvent faire passer dans un même état. Plusieurs conditions peuvent être vérifiées qui ne conduisent pas au même état ; dans ce cas, c'est la dernière rencontrée qui l'emporte. Enfin si aucune condition n'est vérifiée, il n'y a alors pas de changement d'état.

### 3 - Dispositions (prises) au changement d'état :

Certaines dispositions sont prises au moment même du changement d'état. Ce sont :

LA	:	La variable LA prend la valeur de L au moment du changement d'état,
LRN	:	La variable LR est forcée à "0" tant que l'on reste dans cet état,
VN	:	La variable V est forcée à "0" tant que l'on reste dans cet état,
CHK, param.	:	param. est pris comme nouvelle valeur pour la constante de raideur du ressort,
CHZ, param.	:	param. est pris comme nouvelle valeur pour la constante de frottement.

On peut résumer le fonctionnement de ce dispositif par un automate d'états (Figure ci-dessous).



**Figure 14**  
Liaison Conditionnelle

La base algorithmique ainsi constituée des trois éléments :

*Masse Ponctuelle,*  
*Liaison Ressort-Frottement,*  
*Liaison Conditionnelle,*

plongée dans le formalisme général représente la substance principale du premier système CORDIS-ANIMA implanté et utilisé pour de nombreuses simulations depuis environ dix ans.

Pour caractériser entièrement ce premier état de CORDIS-ANIMA, nous devons encore y ajouter toutefois les quelques compléments pratiques ci-dessous.

### 3 - Modules Dégénérés et Modules Intégrés

Les trois modules précédents sont des modules génériques, correspondant à des catégories complémentaires. Ce que nous appelons "système", dans CORDIS-ANIMA, est dans les principes et les catégories importantes. Mais cela n'interdit pas une certaine souplesse : en l'occurrence, ici, à condition d'en connaître les comportements (soit expérimentalement, soit par l'analyse mathématique), nous ne nous interdisons pas de mettre en œuvre des variantes de ces éléments. Par exemple, on pourra utiliser d'autres approximations de la vitesse et de l'accélération pour l'algorithme de la Masse...

Mais deux dispositions pratiques nous ont permis dès ce niveau d'enrichir le langage sans complexité supplémentaire : l'adoption de certains modules dits "dégénérés" pour des fonctions spéciales, et l'"intégration" de plusieurs modules en modules macroscopiques.

- Les modules dégénérés ont de particulier non seulement le fait d'être en général incomplets du point de vue de leur communication, mais aussi le fait qu'ils servent une fonction pratique.

Nous en montrons deux exemples simples :

- le module *SOL* :

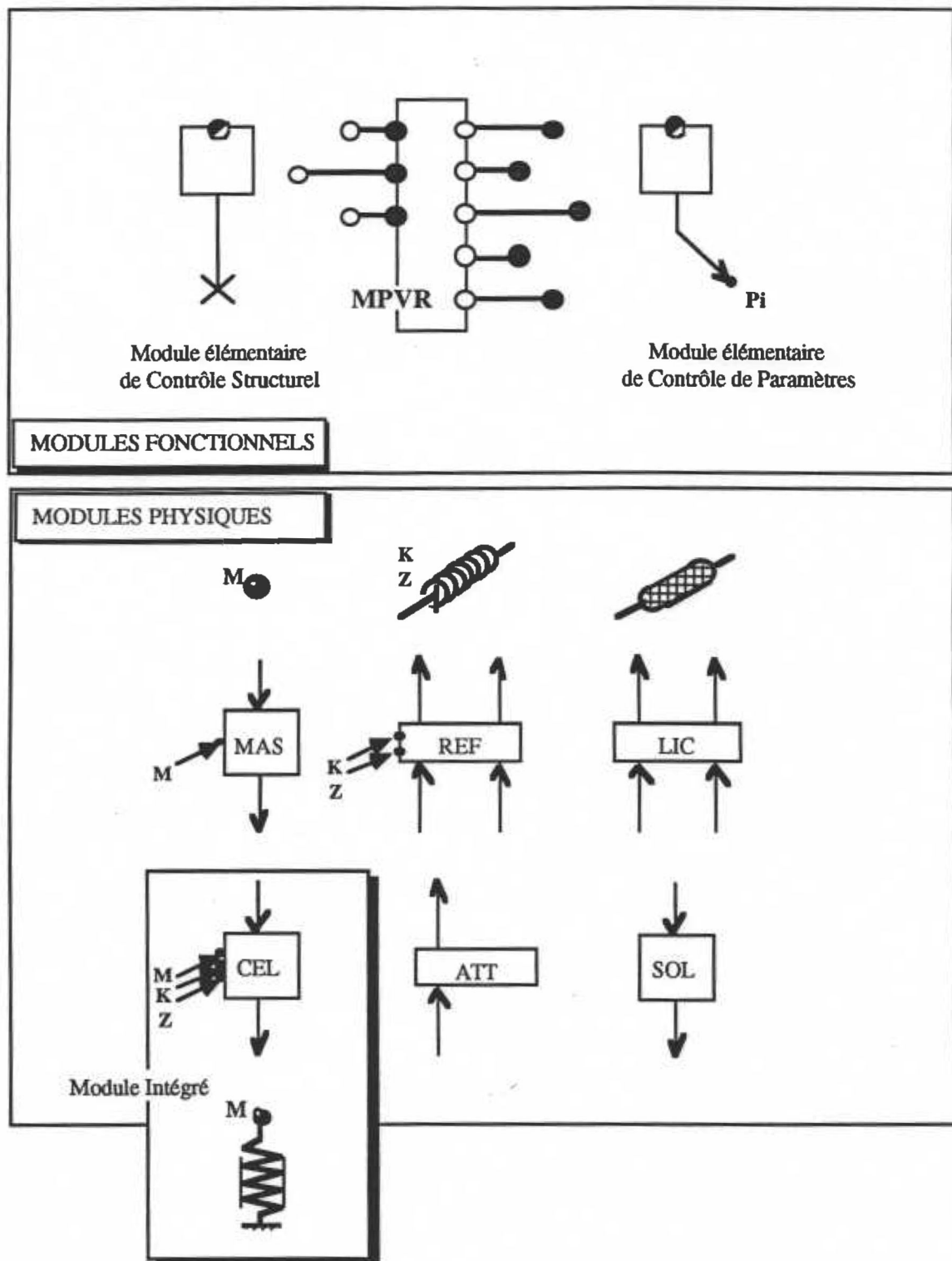
Homogène à un point matériel du point de vue de sa "connectique", il reçoit toutes les valeurs de force que l'on souhaite à son entrée (point M) pour ne jamais fournir autre chose qu'une position obstinément fixe. Cette position, définie auparavant par l'utilisateur est d'ailleurs un paramètre (non contrôlable dynamiquement) et peut signifier la position dans l'espace d'un point d'attache pour un (ou plusieurs) élément(s) ressort-frottement d'une structure quelconque. On peut a priori disposer d'autant de points *SOL* qu'on le souhaite.

- Le module *PES* (Pesanteur) :

Dans le même ordre d'idées, un module de type liaison, dégénéré en ce sens d'une part qu'il n'en n'est qu'une moitié, d'autre part parce qu'à toute valeur de position il répondra invariablement par une valeur de force, permet d'approcher la fonction de la pesanteur. "Approcher" parce que ce module applique une force et non une accélération (ce qui rend la pesanteur dépendante de la masse !), et aussi parce que l'on peut décider si un élément de matière est, ou non, soumis à la pesanteur.

On peut bien évidemment imaginer d'autres modules.

- L'"intégration" de plusieurs modules élémentaires en un seul module macroscopique est une disposition intéressante. En effet, dans les structures rencontrées pratiquement certains éléments composites apparaissent souvent avec insistance et multiplicité et il est alors inutile, voire gênant, de disposer du degré d'atomicité total de nos trois composants fondamentaux. Dans la mesure où, en combinant en un seul algorithme les composants de base, on réalise des économies matérielles (de mémoire, de processus de communication, de complexité structurelle ...), on peut combiner l'économie matérielle et l'économie conceptuelle en proposant des modules "intégrés".



**Figure 15**  
La base algorithmique de Cordis-Anima

### III - EXTENSIONS

#### 1 - La Liaison Conditionnelle Généralisée

Dans la Liaison Conditionnelle du premier système CORDIS, les *états* de la liaison sont caractérisés par la valeur spécifique des paramètres d'une liaison visco-élastique dont l'algorithme est permanent. Ce procédé est très économique, pour une grande généralité.

Ce principe même d'économie est cependant contredit par un autre facteur, en particulier dans le cadre de l'extension quantitative des simulations. Dans les objets complexes, à grand nombre de points matériels et à forte connectivité interne, le nombre de liaisons augmente comme le carré du nombre de points. Si, comme dans certaines applications de l'image animée (simulation de fluides à partir d'un grand nombre de particules), les interactions entre points sont "conditionnelles", un gain considérable peut être obtenu, dans le cas où ces points ne sont pas suffisamment proches pour être en interaction mécanique, en remplaçant le calcul des distances entre points par un calcul de proximité (c'est-à-dire en fait un calcul de distance très simplifié). Pour cela, il est nécessaire de pouvoir changer d'algorithme lorsque l'on change d'état dans la liaison conditionnelle.

Cette raison et un certain nombre d'autres issues de l'expérience nous ont conduit à étendre le principe de la liaison conditionnelle en définissant les états non plus seulement par les paramètres d'une simple liaison visco-élastique, mais par des algorithmes spécifiques.

Cette extension reste toutefois prudemment mesurée : les algorithmes en questions sont des algorithmes compatibles avec la notion d'élément de liaison définie dans la première partie, et le nombre d'états possibles pour une même liaison conditionnelle reste faible (de l'ordre de 2 ou 3).

Enfin, comme algorithme typique susceptible de caractériser l'un de ces états, signalons la *fonctionnelle linéaire par morceaux* : Comme son nom l'indique, elle effectue le calcul de la force de liaison à partir d'une fonction (de la distance entre les points) linéaire par morceaux. Une application simple de cet algorithme est le calcul de l'interaction, dans ANIMA en particulier, entre des particules sphériques libres dans l'espace, mais non pénétrables mutuellement.

#### 2 - Intégrations larges

Nous avons déjà introduit le principe de l'intégration de plusieurs modules élémentaires en un seul "macro-module". Une des évolutions importantes dans la pratique de CORDIS-ANIMA est le recours de plus en plus systématique à des modules à "grande intégration", plus précisément des modules à grand nombre d'éléments identiques communiquant avec leur environnement par un faible nombre de points.

Ceci exploite une propriété structurelle courante des objets vibrants ou mobiles et déformables : ils sont en général décomposables en un petit nombre de macro-constituants. Ces derniers sont le siège individuellement de phénomènes à variable d'état de grande dimensionnalité, mais ils sont liés entre-eux par des connexions à faible dimensionnalité. Citons à titre d'exemple le cas du marteau et de la corde de piano ou de la corde et de la caisse de résonance.



L'exemple le plus simple est l'oscillateur ou *Cellule CORDIS*, combinant une masse, un ressort et un frottement, les deux derniers étant parallèles et reliés d'une part à la masse, d'autre part à un point fixe. D'autres intégrations sont naturellement envisageables : la ligne (ou corde) constituée d'une chaîne de masses (ou de cellules), liées par des ressorts-frottements, la surface, le groupe de cellules ... Nous verrons plus loin, avec le *module modal*, un exemple utilisant un groupe de cellules.

*Remarque :*

- La nécessité pour certaines situations de l'atomicité des modules de base et l'intérêt de l'intégration sont deux contraintes contradictoires. C'est la raison pour laquelle nous avons introduit cette possibilité d'ajouter, de manière libre et ouverte, aux côtés des modules de base, des *modules intégrés*. Par définition, ces modules n'échappent pas à la logique du système car leur fonction peut se décrire à partir des éléments de base. Ils supposent cependant une intervention hors langage, une méta-programmation, par un "méta-opérateur" qui doit être informé des particularités matérielles des machines et qui doit se plonger dans une écriture des équations et des algorithmes.

Le tableau de la page suivante (Figure 15) synthétise les fonctions de ce premier état de CORDIS-ANIMA.

#### 4 - Conclusion de la deuxième partie

Ce que nous avons posé, dans tout ce qui précède, est le centre du système CORDIS-ANIMA ; avec une métaphore plus pittoresque on pourrait dire les racines et une partie du tronc. Plusieurs (im)plantations successives de ces racines dans divers "terrains" (essentiellement plusieurs configurations de machines) ont permis d'expérimenter et d'optimiser leurs propriétés. Dans le même temps la "personnalité" du système s'est précisée. Personnalité en effet car CORDIS-ANIMA vise (cela va de soi !) un objectif universel, mais part d'un point singulier et emprunte un chemin réel, nécessairement contingent.

Alors le plus important maintenant n'est pas dans l'idée d'un système final, absolu par sa complétude et sa cohérence, mais dans la vigueur et la potentialité de ces racines. C'est dire qu'une autre part de CORDIS-ANIMA est constituée de ce qui "pousse" à partir des racines et du tronc : les extensions, qui s'éloignent du centre mais ne pourraient exister sans celui-ci. Par principe, la description de ces extensions ne peut être exhaustive. Par contre, on peut signaler les directions privilégiées. Dans la partie qui suit, nous esquissons quelques unes de ces directions.

L'économie algorithmique peut tirer un parti non négligeable de telles particularités, notamment au travers de la vectorisation des calculs.

Parmi les modules à forte intégration mis en œuvre depuis plusieurs années, on peut évoquer entre autres la *corde modale*, l'*agglomérat* (spécifique d'ANIMA), la "*chambre modale*"...

La corde modale [11] préfigure la "chambre modale" qui la généralise et se décrit de la manière suivante :

#### *La chambre modale (ou module modal)*

[voir P. DJOHARIAN, dans ce volume].

Elle est constituée de  $n$  *cellules* CORDIS indépendantes. Chacune de ces cellules a une masse égale à 1 et se caractérise par ses paramètres de raideur et de viscosité. On peut appeler ces cellules *les modes*.

A cet ensemble est associé un module (fonctionnel) MPVR (cf. première partie) caractérisé par une matrice de changement de base. Cette matrice à  $m$  colonnes  $n$  lignes permet, en entrée de distribuer  $m$  signaux de force sur les  $n$  entrées de force indépendantes des cellules, en sortie, de "mixer" les  $n$  sorties de position indépendantes des cellules sur  $m$  sorties de position.

Ce qui est caractéristique est naturellement la valeur des coefficients de la matrice. Ces derniers sont calculés à partir de la définition d'un objet auquel le module modal sera finalement équivalent : un réseau topologique de type CORDIS de  $n$  masses et  $m$  points de communication.

Le module modal complet comporte  $n \times m$  paramètres pour la matrice,  $2 \times n$  paramètres pour les cellules. Les paramètres des cellules sont contrôlables dynamiquement. Le nombre de cellules n'est pas contrôlable dynamiquement.

Des descriptions plus détaillées sont données par ailleurs (voir P. DJOHARIAN, et J.L. FLORENS, ce volume).

L'intérêt de ce module réside dans plusieurs points :

- Le nombre de calculs relatifs aux liaisons se limite à  $n$  (le nombre de modes), alors que pour un réseau en général, il est compris entre  $n$  et  $n(n-1)/2$ .
- Le nombre de communications avec le reste de la structure ( $m$ ) peut être très faible par rapport au nombre ( $n$ ) de modes.
- Il réalise, pour la structure vibrante, une correspondance entre approche causale (la description topologique du réseau) et approche perceptuelle (la description modale, proche de la description fréquentielle).

L'agglomérat est un terme générique correspondant à une catégorie de modules développés en particulier dans ANIMA. Nous ne donnerons ici, à titre de simple évocation pour laisser la description spécifique de ANIMA à une place plus appropriée, qu'une description succincte du cas le plus simple. Pour plus de détails, nous renvoyons à la bibliographie [4], [6], [9], [12], [16], [18].

Dans sa version la plus élémentaire, l'agglomérat est un ensemble de  $n$  particules toutes en interactions les unes avec les autres. L'homogénéité d'un tel *objet* autorise une simplification très pertinente de l'algorithme. Celui-ci se résume à deux macro-modules couplés. L'un des deux, vectorisé sur l'ensemble des particules (masses ponctuelles), calcule toutes leurs positions, l'autre, vectorisé sur les  $n(n-1)/2$  liaisons en cause, calcule selon le même algorithme, toutes les interactions.

### 3 - Les modules "ad hoc"

Un des aspects du processus de construction de CORDIS-ANIMA est la relation dialectique qui s'établit entre la formalisation et la mise en œuvre du matériau algorithmique proposé par les machines. Une vision trop unilatérale de cette relation stériliserait le processus, aussi, parfois, est-il nécessaire et fécond d'abandonner temporairement le guide formel pour développer une économie algorithmique plus en amont de celui-ci.

Le module "ad hoc" fait partie de la démarche CORDIS-ANIMA dans la mesure où il est une "négociation" bien comprise de la frontière entre ce qui est orthodoxe et ce qui ne l'est pas. On peut ainsi parler d'une manière formelle de ce qui échappe à la formalisation.

Le module ad hoc est un module dans lequel tout est permis du point de vue algorithmique, c'est-à-dire que l'on peut, si cela présente un intérêt, s'éloigner des notions de point matériel, de liaison mécanique etc. Il y a simplement les restrictions suivantes : sa fonction globale doit rester mécanique, il doit correspondre à un modèle physique et en particulier présenter tous les aspects de "conformité sociale" avec les modules CORDIS-ANIMA en général : se présenter selon un certain nombre de points de communication de type M ou L, avec d'éventuels paramètres de contrôle.

Grâce à cette disposition, CORDIS-ANIMA peut intégrer dans son espace d'expérimentation des démarches de modélisation physique de diverses natures comme celle de la ligne à retard de Karplus-Strong (Karplus-Strong 1983) ou des guides d'ondes de J.O. Smith (Jaffe, Smith, 1983). Nous avons nous-mêmes développé par ce canal des modèles particuliers de frottement d'archet [19] qui se sont avérés très pertinents.

Enfin, il convient de ne pas confondre cette ouverture avec le principe de l'intégration dans lequel les éléments intégrés restent descriptibles en termes de *masses, ressorts, frottements et liaisons conditionnelles* élémentaires.

Il ne faut pas plus confondre cette catégorie spéciale de modules avec celle que nous introduisons maintenant.

#### 4 - Les modules "extra-physiques"

Toute démarche n'a d'identité véritable que si elle est à même de dialoguer avec ce qu'elle n'est pas. Il s'agit d'une nouvelle expression d'un certain principe d'ouverture des systèmes que nous avons déjà adopté de plusieurs façons dans ce qui précède. Ici, nous allons un peu plus loin en ménageant cette autre ouverture aux modules "extra-physiques" c'est-à-dire à la possibilité d'intégrer à CORDIS-ANIMA des modules pour lesquels on ne cherche a priori aucune référence dans les processus physiques réels.

Il faut comprendre cette démarche selon deux niveaux distincts : le niveau pragmatique ou technique et le niveau "esthétique". Les guillemets sur le dernier terme signifient qu'il faut le prendre dans le sens large de toute la démarche relative au processus de création.

A ce niveau esthétique, ce que nous introduisons explicitement est une des formes possibles d'articulation entre référence et artifice. Les modules extra-physiques - que nous désignons avec ce nom exotique pour bien les distinguer des *modules fonctionnels* de la base et pour dire surtout qu'ils ne sont pas physiques sans avoir à dire ce qu'ils sont vraiment - doivent précisément permettre d'introduire dans le corps même de l'objet central, des processus dynamiques, des principes générateurs de forme répondant d'une autre logique que celle de la mécanique, de l'instrumentalité physique du monde. Peuvent s'y engouffrer, après toute la retenue formalisante et systématisante de notre première partie, tous les laissés pour compte de la modélisation acoustique par exemple...

*Cependant !*

Cependant, tout ce qui précède n'aurait servi à rien si ce n'était que pour lâcher la bride en fin de course, au nom de la généralité. On ne peut faire un système que de généralité, c'est équivalent à pas de système du tout.

En fait, la frontière reste relativement bien tracée : ce que nous abandonnons dans les modules extra-physiques n'est en fin de compte que la cohérence énergétique propre aux objets physiques. Les modules en question doivent sagement rester à l'intérieur d'une enveloppe de *points de communication* et les variables de communication, force et déplacement doivent permettre l'échange avec des modules orthodoxes. Passées la frontière, du côté interne au module, ces variables acquièrent leur autonomie, peuvent devenir des signaux de contrôle de la fréquence, des amplitudes, ...

Le seul risque encouru est alors, en effet, que selon le processus modélisé par un tel module, des incohérences énergétiques surgissent, correspondant à des sources implicites où des pertes intempestives.

On peut alors imaginer, là où cela présenterait un intérêt, d'intégrer un paradigme tel que celui de MUSIC V (pour ne prendre que celui-ci parce qu'il est le plus ancien et le plus "populaire") dans CORDIS-ANIMA et, qui plus est, de réaliser une liaison intime entre la modélisation physique et la modélisation acoustique que constitue ce système. Remarquons au passage que le rôle métaphorique de l'un par rapport à l'autre pourrait être aussi bien joué dans un sens que dans l'autre. Par exemple, un signal de contrôle produit par un module de type oscillateur créerait une forme de modification non physique de la raideur d'un élément vibrant de CORDIS, produisant une forme musicale élémentaire. Mais, en sens inverse, un processus très physique comme le rebond amorti d'une balle sur une surface dure, pourrait

contrôler l'évolution spectrale d'un son réalisé à partir d'un modèle complexe de filtrage ou de synthèse additive.

On voit bien que cette ouverture n'est pas un brusque laxisme par rapport à tenue systématique du début. En effet, c'est par le fait que ces modules extra-physiques sont tenus de se lover dans le format orthodoxe des modules que l'on jugule tout risque d'hémorragie. Cette attitude permet d'identifier rigoureusement l'emplacement des îlots non physiques et, par la même, d'une part de circonscrire leurs effets indésirables, d'autre part d'exploiter leurs effets intéressants.

Remarquons qu'en termes concrets ce que l'on vient de dire ne coûte rien. Il n'y a pas, à ce jour, dans nos propres applications, de module extra-physique effectivement utilisé. Leur développement est à la charge d'un "super-user" de CORDIS-ANIMA. Toutefois, bien qu'il ne s'agisse que d'un principe, nous tenons à l'associer aux clauses de définition même de CORDIS-ANIMA.

Quant au niveau pragmatique et technique annoncé, on peut l'évoquer avec cette remarque finale :

Le plongement d'un système comme MUSIC V dans CORDIS-ANIMA est possible alors que l'inverse ne l'est pas. Plus précisément, la forme structurelle sous-jacente au second est plus générale que celle qui sous-tend le premier et de nombreux systèmes ou méthodes de synthèse développés après lui puisqu'il s'agit d'un réseau et de communications bi-directionnelles au niveau le plus intime. Une dégénérescence du réseau en arbre et des communications en communications unilatérales nous donnent les formes structurelles propres à ces systèmes antérieurs.

Mais nous ne faisons là qu'énoncer une banalité sans mérite car ce qui est en cause ici n'est pas propre à notre problème, mais correspond à la tendance actuelle de l'évolution des machines vers le parallélisme et de la pensée informatique vers la programmation parallèle. Toutefois, peut-on alors dire que CORDIS-ANIMA correspond à cette tentative de penser les processus générateurs et l'interaction avec le sujet en termes de communication bilatérale à tous les niveaux, donc d'une certaine manière de parallélisme.

## CONCLUSION

Cette présentation de CORDIS-ANIMA ne comporte aucune illustration par des exemples d'application. C'est un choix délibéré car pour ne pas allonger démesurément le texte nous aurions dû nous contenter de quelques exemples seulement, ce qui ne donnerait pas une image juste de la généralité. Nous renvoyons à d'autres présentations pour cette illustration. Par contre, nous pouvons en conclusion formuler quelques commentaires généraux sur les propriétés fondamentales de ce système dans le cadre d'une problématique de création.

CORDIS-ANIMA fait de l'ordinateur, doté des transducteurs sensoriels, un moyen de représentation, ce qui constitue la première fonction nécessaire à la création. Partant de là deux choses sont essentielles à la démarche sous-jacente : d'une part le fait que le processus



de création soit considéré comme une situation dynamique de dialogue à de multiples niveaux avec le moyen de représentation lui-même, d'autre part le choix de l'objet central de cette représentation. *Ici, cet objet central, c'est l'objet physique.*

Il ne s'agit pas d'une régression vers un primitivisme sensori-moteur devant se substituer aux représentations abstraites dont l'intelligence contemporaine aime à se montrer capable.

Penser cela, c'est ne pas avoir compris toute la signification et toute la portée de cette situation sans antécédent : l'ordinateur n'est pas l'instrument, il est, un ordre plus loin, un moyen de représentation de l'instrument, un "instrument d'instruments". L'objet physique et la représentation de l'objet physique sont deux choses distinctes. C'est par la seconde que le premier acquiert un nouvel ordre de réalité inexistant jusque là. Ce nouvel ordre de réalité est celui qui est compatible précisément avec les sommets d'abstraction atteints aujourd'hui grâce à l'informatique.

L'objet physique constitue donc le centre du processus, le principe de permanence caractéristique du paradigme.

CORDIS-ANIMA, dans sa première fonction, permet de le *créer*.

Cette permanence est à son tour constituée d'une permanence plus fondamentale, non plus temporelle, mais structurelle : ce que nous avons appelé la "matière linéaire". De ce substrat homogène part alors un faisceau de variations possibles, au sens d'introduction de la variété :

Tout d'abord celui des non-linéarités et discontinuités structurelles, introduites par le concepteur, mais intrinsèque à l'objet. C'est le rôle des modules de variation dynamique lorsqu'ils sont contrôlés par un autre module de l'objet lui-même. C'est le rôle également des liaisons conditionnelles...

Il y a ensuite celui de la variation résultant du dialogue direct entre l'objet construit et l'opérateur. C'est la forme la plus intime du dialogue, celle qui exploite a priori toute la capacité communicationnelle de l'homme par le moyen de son *geste*. Les points de communication physique avec l'extérieur, les modules de variation dynamique structurelle, paramétrique,... permettent de chercher l'équilibre optimal entre la permanence de l'instrument et la variation temporelle dont le geste est capable.

Il y a enfin toute la variation possible dans le cadre formel général, à cet "hyper niveau" des modèles, entre la référence au physique et ce que nous avons appelé l'"extra-physique", précisément sans autre précision.

CORDIS-ANIMA est, comme on l'a annoncé en introduction, un système de modélisation et de simulation des objets du monde physique, c'est un outil de modélisation, un outil pour la simulation, un système, un langage... C'est enfin un "état d'esprit".



- [10] CADOZ C., FLORENS J. L., LUCIANI A. 1985. "Procédé et dispositif pour simuler un objet matériel et ses interactions avec un utilisateur" Brevet Français. Déposé le 14 / 06 / 85. n° 85 09420.
- [11] FLORENS J. L., LUCIANI A., CADOZ C. 1986. "Optimized real-time simulation of objects for musical synthesis and animated images synthesis" *International Computer Music Conference*. La Haye 1986.
- [12] RAZAFINDRAKOTO A. 1986. "Langage d'objets pour l'animation, implantation temps réel" *Thèse d'Université*. Grenoble 1986.
- [13] CADOZ C. 1986. "Langage pour la définition et la simulation d'objets instrumentaux, Informatique Musique Image Animée" *Rapport ACROE*, III.2. Grenoble 1986.
- [14] CADOZ C. 1988. "Informatique et Outil de Création Musicale" *Revue Marsyas* n°7. Institut de Pédagogie Musicale et Chorégraphique. La Villette, pp. 18-29. Paris 1988.
- [15] CADOZ C., LUCIANI A., FLORENS J. L. 1989. "Responsive Input Devices and Sound Synthesis by Simulation of Instrumental Mechanisms : The Cordis System" in *Music Machine* Curtis Roads Ed. MIT Press, Cambridge Mass. 1989.
- [16] JIMENEZ S., LUCIANI A. 1989. "An unified view of multitude behaviour, flexibility, plasticities and fractures" *Rapport de recherche "Lifia 104-Acroe 89-15"*. Grenoble Nov. 1989.
- [17] JIMENEZ S. 1989. "Modélisation et simulation d'objets volumiques déformables complexes" *DEA Informatique*, I.N.P.G. Grenoble 1989.
- [18] LUCIANI A. 1989. "Physical Models in animation, Towards a modular and instrumental approach" *Rapport de recherche "Lifia 103-Acroe 89-14"*. Grenoble Nov. 1989.
- [19] FLORENS J. L. 1990. "Modèles et simulations en temps réel de cordes frottées" *1er Congrès Français d'Acoustique*. SFA, Editions de Physique, Lyon Avril 1990.
- [20] FLORENS J. L., CADOZ C. 1990. "Modular Modelisation and Simulation of the Instrument" *International Computer Music Conference*. Glasgow 1990.
- [21] FLORENS J. L., CADOZ C. 1991. "The physical Model, Modelisation and Similation Systems of the Instrumental Universe" in *Representations of Musical Signals*. G. De Poli, A. Picciali, C. Roads, Ed. MIT Press, 1991.
- [22] CADOZ C. 1991. "Timbre et Causalité" in *Le timbre, métaphore pour la composition* Barrière J. B., Paris, Christian Bourgois 1991.